Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Miha Oblak

Strategija hierarhije hrambe podatkov v zdravstvu

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

prof. dr. Miha Mraz
MENTOR

Ljubljana, 2016

Tematika naloge:

Kandidat naj v svojem delu analizira možne strategije hrambe zdravstvenih podatkov z vidika potreb regijske bolnišnice v slovenskem okolju. Pri tem morata biti dosežena predvidena normativa razpoložljivosti in zanesljivosti. V predlagani splošni rešitvi naj kandidat izpostavi aktualne tehnološke možnosti na področjih hierarhije hrambe podatkov, njihove replikacije, kompresije in deduplikacije ter redundančne hrambe.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani izjavljam, da sem avtor dela, da slednje ne vsebuje materiala, ki bi ga kdorkoli predhodno že objavil ali oddal v obravnavo za pridobitev naziva na univerzi ali drugem visokošolskem zavodu, razen v primerih kjer so navedeni viri.
S svojim podpisom zagotavljam, da:

- sem delo izdelal samostojno pod mentorstvom prof. dr. Mihe Mraza,
- so elektronska oblika dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko in
- soglašam z javno objavo elektronske oblike dela v zbirki "Dela FRI".

— Miha Oblak, Ljubljana, maj 2016.

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Miha Oblak

Strategija hierarhije hrambe podatkov v zdravstvu

POVZETEK

Visoka razpoložljivost podatkov v zdravstvu je nujna, saj na bi bili po vpeljavi projekta e-Zdravje ključni podatki dostopni vsem zdravstvenim ustanovam. Ti podatki so pogoj za hitro in uspešno obravnavo pacienta v katerikoli zdravstveni ustanovi. V Sloveniji je zavedanje o pomenu visoke razpoložljivosti podatkov in zanesljivosti dostopa v hitrem razvoju. S staranjem prebivalstva in ob povečanem številu obravnav z modernimi tehnologijami se hitro povečuje količina podatkov o posameznem pacientu. Aktualnih podatkov je relativno malo, zato je vse pomembnejše vprašanje, kako čim ceneje hraniti starejše podatke, katerih hrambo narekuje zakon.

V diplomski nalogi so predstavljene tehnologije za povečevanje razpoložljivosti podatkov na lokalni ravni in možnosti za replikacijo podatkov na rezervno lokacijo. Podani so tudi načini za brezizgubno zmanjševanje količine podatkov na različnih nivojih informacijskega sistema. Z opisom prednosti in slabosti posamezne možnosti lahko vsak oceni, kateri način bi bil za njegov poslovni proces najprimernejši.

Za potrebe fiktivnega naročnika je predstavljen postopek izbora ustreznih tehnologij glede na izdelan načrt neprekinjenega poslovanja in definirane zahteve delovnega procesa, pa tudi primer izračuna potrebne opreme za vpeljavo predlagane rešitve. V izračun je vključena optimizacija z izrabo obstoječih virov.

Ključne besede: razpoložljivost podatkov v zdravstvu, replikacija, kompresija, deduplikacija, upravljanje hierarhične hrambe podatkov, načrt neprekinjenega poslovanja

University of Ljubljana
Faculty of Computer and Information Science

Miha Oblak

Hierarchical Storage Management Strategy in Health Care

ABSTRACT

High availability of data in healthcare is essential, since the introduction of e-Health project patient key information should be available to all health institutions. These data are condition to fast and efficient patient care in any healthcare institution. Awareness of the importance of high data availability and reliability is fast developing in Slovenia. With aging population and increasing number of treatments with modern technologies, amount of each patient data is rapidly increasing. Current data amount is relatively low but question, how to financial effective store older data required by regulatives, is becoming important.

In this thesis are presented technologies to increase data availability on local level and also possibilities for data replication to remote location. We also present methods for lossless data amount reduction on different levels of the information system. By describing the advantages and disadvantages of each option each one can assess which way would be the most appropriate for his business process.

For the purposes of the notional principal the selection process of appropriate technologies in relation to the constructed business continuity plan and the defined requirements of the working process is presented, as well as the example of the calculation of necessary equipment in order to introduce the suggested solution. The calculation includes optimization with reuse of existing resources.

Key words: Data availability in healthcare, replication, compression, deduplication, hierarchical storage management, business continuity plan

ZAHVALA

Zahvala ob zaključku tega dela gre mentorju prof. dr. Mihi Mrazu, ki mi je s hitrim odzivom in zelo korektnim odnosom pomagal pri končanju študija in me usmerjal pri pisanju diplomskega dela.

Hvala ženi in otrokom za vse sokove in prigrizke, pa tudi Bernikovim, pri katerih so bili v času pisanja dokaj stalni gostje.

Zahvalil bi se tudi sodelavcem za nadomeščanje, šefom za plačo in prof. dr. Dušanu Kodeku za zadnjo desetko.

— Miha Oblak, Ljubljana, maj 2016.

SEZNAM UPORABLJENIH KRATIC

Kratica	Pomen
BCP	Načrt neprekinjenega poslovanja (angl. *Business Continuity Plan*)
EMS	Evropska potresna lestvica
FC	Protokol za komunikacijo po SAN omrežjih (angl. *Fibre Channel*)
HSM	Upravljanje hierarhične hrambe podatkov (angl. *Hierarchical Storage Management*)
ILM	Upravljanje z življenjskim ciklom informacije (angl. *Information Lifecycle Management*)
LTO	Format magnetnih trakov (angl. *Linear Tape Open*)
LUN	Virtualni disk na diskovnem sistemu (angl. *Logical UNit*)
LVM	(angl. *Logical Volume Manager*)
LV	Particija na disku (angl. *Logical Volume*)
LTFS	Datotečni sistem na magnetnih trakovih (angl. *Linear Tape File System*)
RAID	Redundančno polje diskov (angl. *Redundant Array of Independent Disks*)
RDMA	Direkten dostop do pomnilnika oddaljenega strežnika (angl. *Remote Direct Memory Access*)
RPO	Najbolj oddaljen čas, za katerega imamo konsistentne podatke v primeru incidenta (angl. *Recovery Point Objective*)
RTO	Čas, v katerem je potrebno vzpostaviti poslovni proces v primeru incidenta (angl. *Recovery Time Objective*)
SAN	Diskovno omrežje (angl. *Storage Area Network*)
SVC	IBM San Volume Controler, strežnik za diskovno virtualizacijo
TCO	Skupni stroški lastništva (angl. *Total Cost of Ownership*)

KAZALO

1 Uvod

Evropska unija je že leta 2000 sprejela osnovne smernice za e-Evropo, projekt, ki bi digitalno povezal države članice na vseh področjih. Na področju zdravstva je bil v Sloveniji leta 2006 predviden začetek projekta e-Zdravje. Do danes je vpeljanih 17 rešitev, ki omogočajo veliko prednosti za vse udeležence v zdravstvu - pacientom dostop do pravih informacij in e-storitev, zdravstvenim delavcem pa celostni vpogled v zdravstveno stanje posameznika, ne glede na kraj in čas predhodnih obravnav, olajšano komunikacijo med specialisti, hitro in učinkovito izmenjavo podatkov med urgenco in policijo, itd. Ena izmed teh rešitev je tudi postavitev interoperabilne hrbtenice [1], ki je primer federativnega pristopa do podatkov. Vsak podatek se hrani na mestu, kjer je nastal, centralna točka pa ima pregled nad lokacijami ključnih podatkov [2]. Posamezni avtorizirani podatki se hranijo na certificiranih točkah izvajalcev zdravstvene dejavnosti in so na voljo tudi ostalim zdravstvenim delavcem na vseh ravneh zdravstvene oskrbe, ki sodelujejo v postopku zdravljenja. Zaradi takega načina izmenjave podatkov morajo biti podatki razpoložljivi praktično ves čas. Vsak uporabnik interoperabilne hrbtenice mora sam poskrbeti za razpoložljivost svojih podatkov. V regijskih bolnišnicah, kjer je več obravnav pacientov,

so razpoložljivi podatki pomemben člen v poslovnem procesu. Ob hkratnem izvajanju raziskovalne in izobraževalne dejavnosti se pomen razpoložljivosti samo še povečuje.

V 2. poglavju diplomskega dela je predstavljeno okolje fiktivnega naročnika projekta in izzivi, s katerimi se spopada vodstvo. Izzivi so prisotni na področjih zagotavljanja razpoložljivosti podatkov, njihovega učinkovitega hranjenja in obvladovanja informacijskega sistema s čim manjšimi skupnimi stroški lastništva.

3. poglavje je namenjeno predstavitvi možnosti za obvladovanje velikih količin podatkov ob hkratnem zagotavljanju relativno kratkih časov za nadaljevanje poslovanja v primeru kritičnega dogodka, naj bo to okvara podatkovnega skladišča, nesreča (požar, poplava, itd.) ali pa namerno uničenje. Današnje tehnologije in možnosti povezljivosti omogočajo prenos velikih količin podatkov tudi na daljše razdalje, zato imamo možnost zadostiti različnim potrebam in zahtevam poslovnih procesov po omogočanju dostopa do podatkov. Z naraščanjem števila obravnav pacientov in življenjske dobe se povečuje količina dokumentacije, ki jo je potrebno hraniti na daljše časovno obdobje. Obenem se v dokumentacijo prilaga tudi čedalje več video in grafičnih zapisov. Količina podatkov pogosto presega zmožnosti hranjenja v aktivnem delu, hkrati pa zahteva dostop do podatkov v relativno kratkem času in predvsem brez človeškega posredovanja.

Optimizirana rešitev problema fiktivnega naročnika je predstavljena v 4. poglavju. Izdelan je okvirni načrt za neprekinjeno poslovanje in izračunane potrebne komponente, ki zadostijo zahtevam. Tak izračun je potrebno narediti za vsakega naročnika posebej, prav tako tudi načrt za neprekinjeno poslovanje. Geografska raznolikost v Sloveniji namreč predstavlja zelo različne verjetnosti za posamezen kritičen dogodek (poplava, potres, itd.) s samo lokacijo in vplivom na vreme. Vplivni so še demografska raznolikost, stopnja brezposelnosti, BDP na področju zdravstvene ustanove, itd.

Zagotavljanje zanesljivosti in visoke razpoložljivosti podatkov je ob maloštevilčnih podpornih ekipah zahtevno opravilo, saj zahteva poglobljena znanja na vseh nivojih informacijskega sistema.

2 Opis problema

Predpostavimo, da je fiktivni naročnik projekta teme pričujoče diplomske naloge regijska bolnišnica, ki opravlja zdravstveno dejavnost na primarni, sekundarni in terciarni ravni ter izobraževalno in raziskovalno dejavnost [3]. Vodstvo zdravstvene ustanove je postavljeno pred izziv, kako zagotoviti učinkovito hrambo podatkov iz performančnega in finančnega stališča, hkrati pa zadostiti zahtevam narave poslovnega procesa po hitri vzpostavitvi poslovanja v primeru kritične napake na primarni lokaciji hrambe podatkov in informacijskega sistema. Obenem vodstvo želi zaradi lažjega obvladovanja kritičnih situacij z rešitvijo nadaljevati proces centralizacije informacijskih storitev.

Izraz 'informacijski sistem' v tem primeru predstavlja strežniško in omrežno infrastrukturo nadgrajeno z aplikacijami, s katerimi informacijska služba zagotavlja storitve (zapis, posredovanje in varovanje podatkov) v notranjem omrežju in ne vključuje strojne opreme uporabnikov in merilnih naprav.

2.1 Opis naročnikovega sistema

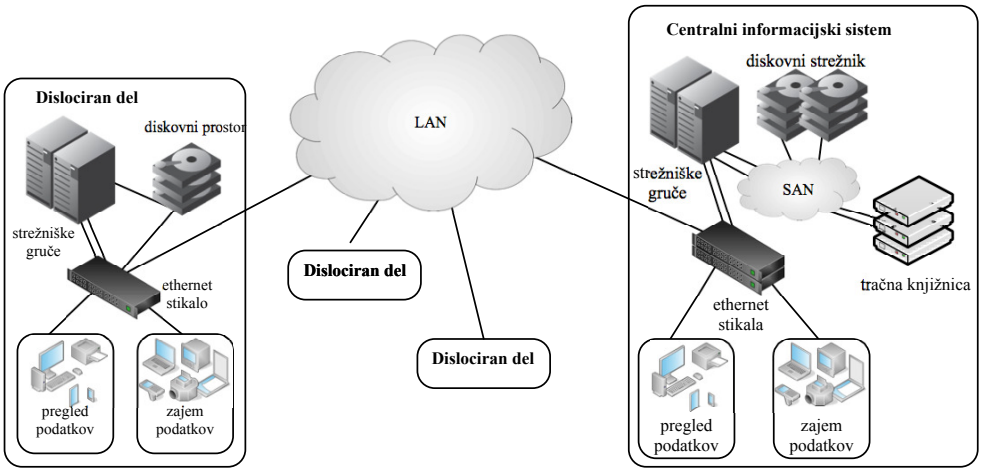
Hipotetična ustanova je organizirana v 15 strokovnih klinik in 4 oddelke za spremljajoče aktivnosti. Organizacijske enote so nameščene v več zgradb znotraj kampusa, kjer sta tudi izobraževalni in raziskovalni center. Informacijski sistem je delno centraliziran. Devet klinik in podporne službe uporabljajo centralni informacijski sistem, ostale klinike pa uporabljajo individualne rešitve. Enotna IT služba izvaja podporo za vse. Centralni informacijski sistem je postavljen v poslovni stavbi v prostoru, kjer je zagotovljena zaščita pred ognjem, vodo in pregrevanjem, prostor pa je tudi tehnično ustrezno varovan v skladu z Zakonom o varstvu osebnih podatkov. Dislocirani deli informacijskega sistema so v klimatiziranih prostorih in so v postopku integriranja v centralni sistem, s katerim so trenutno povezani preko ethernet omrežja in so del enotnega LAN omrežja. Interoperabilnost med sistemi znotraj ustanove je zagotovljena na aplikacijskem nivoju, podobno kot je zagotovljena interoperabilnost z ostalimi izvajalci zdravstvenih storitev, ki so vključeni v interoperabilno hrbtenico [4] preko omrežja zNET [5].

Podatki se hranijo na več diskovnih sistemih, kjer so zapisani na RAID poljih. Varnostna kopija podatkov se dela na magnetne trakove, ki se enkrat tedensko prenašajo v tehnično varovan ognjevaren prostor v banki na nasprotni strani ulice.

Slika 2.1 prikazuje obstoječo arhitekturo informacijskega sistema. Uslužbenci dostopajo do podatkov, jih vpisujejo in urejajo preko aplikacij na prenosnih (tablice, čitalci bar kode) in stacionarnih napravah (delovne postaje, merilne naprave, rentgeni, itd.) preko tankih odjemalcev in direktno iz merilnih naprav. Slednje imajo zagotovljen prostor, kamor lahko začasno shranjujejo podatke, če jih ne morejo posredovati strežnikom. Prostora je za približno 4 ure dela.

2.1.1 Količina podatkov

Vse organizacijske enote skupno letno ustvarijo med 12 in 15 TB podatkov, od tega je približno 60% grafičnih. Hranjeni podatki trenutno zasedajo okrog 300TB prostora. Slikovni in video podatki so kompresirani z uporabo brezizgubnih kompresijskih algoritmov, ostali podatki pa niso kompresirani. V podatkovnih bazah je okrog 90TB podatkov. Dnevno se velika večina podatkov ustvari med 7. in 19. uro.



Slika 2.1 Shema informacijske infrastrukture.

2.1.2 Strojna oprema

Strojna oprema je prilagojena zahtevam in priporočilom posameznih aplikacij. Zaradi časovne razlike pri uvajanju informacijske podpore v posameznih sektorjih, je del strojne opreme namenjen uporabi v določenih organizacijskih enotah, del pa je že virtualiziran in si več organizacijskih enot deli isto strojno platformo. Diskovna polja so bila prilagojena zahtevam aplikacij in so v različnih zmogljivostnih in velikostnih razredih, prav tako so od različnih proizvajalcev.

Strežniki so zgrajeni okrog Intel in IBM Power procesorjev. Intel strežniki so virtualizirani z VmWare vSphere, Power strežniki pa s PowerVM Enterprise. Licence za virtualizacijo omogočajo uporabo naslednjih naprednih funkcij za zagotavljanje razpoložljivosti virtualnih strežnikov:

- migracijo virtualnih strežnikov med fizičnimi strežniki med delovanjem,
- zagotavljanje visoke razpoložljivosti virtualnih strežnikov v primeru odpovedi strojne opreme,
- uporabo redundančnega virtualizacijskega okolja na IBM Power strežnikih.

Tračna knjižnica ima vgrajenih 6 LTO-5 pogonov. Tehnologija omogoča bralno-pisalni dostop do medijev LTO-4 in LTO-5, bere pa lahko še iz medijev LTO-3. V uporabi so samo LTO-5 mediji. Tračna knjižnica se uporablja za centralizirano varnostno kopiranje vseh sklopov informacijskega sistema.

2.1.3 Programska oprema

V tem sklopu se bomo omejili na sistemski del programske opreme in aplikacijski del, ki ponuja skladiščenje podatkov. Zaradi zahtev različnih poslovnih aplikacij so v uporabi operacijski sistemi Windows, Linux in AIX. Podatki se zapisujejo na datotečni sistem ali v podatkovne baze. Vse podatkovne baze so IBM DB2, ki tečejo na operacijskem sistemu AIX. Poslovne aplikacije tečejo na operacijskih sistemih Windows in Linux. Za varnostno kopiranje podatkov se uporablja IBM Tivoli Storage Manager (TSM)[1].

2.2 Zahteve delovnega procesa

Delovni proces se izvaja vsak dan. Potrebujejo takojšen dostop do aktualnih podatkov, vendar je narava delovnega procesa taka, da v primeru nekajminutnega izpada le-ta ne trpi.

2.2.1 Razpoložljivost in varovanje podatkov

Napake na strojni opremi v strežniškem delu informacijskega sistema rešujejo s pomočjo

- podvojenega sistema napajanja,
- uporabe RAID polj na diskovnem sistemu,
- uporabo Etherchannela na mrežnem nivoju,
- podvojenega SAN omrežja,
- visoko razpoložljivih strežniških gruč.

Napake pri okvari podatkov rešujejo z varnostnim kopiranjem po naslednjih pravilih:

- datotečni sistem:
  - v nedeljo: polna varnostna kopija,
  - enkrat dnevno: inkrementalna varnostna kopija.
- podatkovne baze:
  - enkrat dnevno: polna varnostna kopija med delovanjem,
  - enkrat na uro: kopija transakcijskih logov.

S trenutno arhitekturo dosegajo 99,99% razpoložljivost podatkov, kar pomeni slabo uro nepredvidenih izpadov letno.

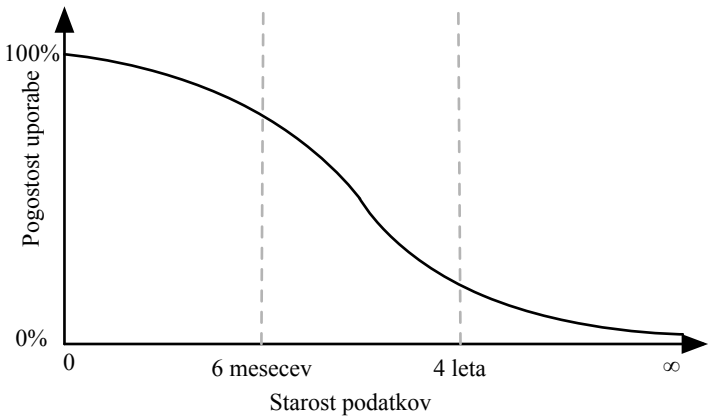
[1] Z verzijo 7.1.3 je preimenovan v Spectrum Protect

2.2.2 Dostopanje do podatkov

Glede na pogostost dostopanja lahko naročnikove podatke razdelimo v tri časovne okvire in sicer v

- podatke, stare do pol leta,
- podatke, stare med pol in štiri leta,
- podatke, starejše od štirih let.

Do podatkov iz prve skupine se dostopa zelo pogosto, do podatkov iz druge skupine redkeje, dostopanja do podatkov, starejših od štirih let pa so zelo redka. Dejansko do teh podatkov dostopajo samo v primeru ponovitve diagnoze in med dolgoročnimi raziskavami. Po približno šestih mesecih od kreiranja podatka pogostost dostopanja strmo pada, po štirih letih pa so dostopanja do podatkov redka, pod 10%. Naročnik je podal oceno pogostosti dostopanj do različno starih podatkov in jo predstavil v grafu na sliki 2.2.



Slika 2.2 Graf pogostosti dostopanja do podatkov.

V tabeli 2.1 so predstavljeni še sprejemljivi dostopni časi do različno starih dokumentov. Dostopni časi izhajajo iz toka delovnega procesa. Podatki iz prve skupine morajo biti dosegljivi praktično vedno. Sprejemljiva motnja za nemoten potek delovnega procesa je, če so dostopni v nekaj minutah[2].

Določen je tudi čas, za katerega so pripravljeni ponovno vnašati podatke, če pride do

[2]S tem je definiran RTO[6]

starost dokumenta	*dostopni čas*	
	običajne razmere	*v primeru izpada*
<6 mesecev	15 sekund	10 minut
6 mesecev <4 leta	1 minuta	30 minut
>4 leta	5 minut	4 ure

Tabela 2.1 Tabela z definiranimi še dopustnimi dostopnimi časi do dokumentov.

napake, ki bi zahtevala restavriranje iz varnostne kopije - ta čas je 30 minut[3]. S trenutno arhitekturo informacijskega sistema zastavljenih ciljev v primeru incidenta nikakor ne morejo dosegati.

2.3 Finančne zahteve

Obstoječa arhitektura omogoča vzdrževanje, osnovno sistemsko administracijo in prvi nivo podpore uporabnikom informacijskega sistema s tremi strokovno usposobljenimi delavci in dvema pomočnikoma. Pričakujemo, da bo po posodobitvi arhitekture informacijskega sistema le-ta ponujal

- stroškovno učinkovitejšo hrambo podatkov,
- razpoložljivost podatkov ob izpadu posameznega diskovenga polja,
- vzpostavitev delovanja storitev ob izpadu primarne lokacije informacijskega sistema v časih, navedenih v tabeli 2.1,
- upravljanje z obstoječimi skrbniki.

Rešitev mora ponujati dovolj kapacitet za naslednjih pet let in možnost enostavne nadgradnje med delovanjem. Za postavitev sekundarne lokacije znotraj ustanove je možno uporabiti katerega izmed obstoječih sistemskih prostorov.

[3] S tem je definiran RPO[6]

3 Možnosti tehnoloških rešitev

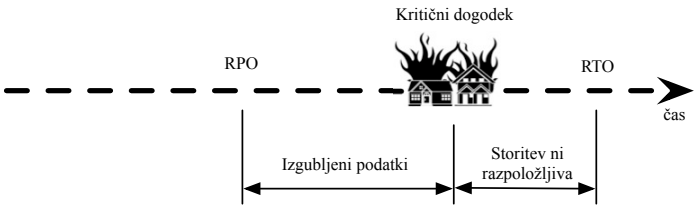
Informatika nam na enostaven način omogoča analizo pojavnosti določenih vzorcev na daljše časovno obdobje, primerjavo simptomov na več vzorcih in tudi večje možnosti za uspešno preventivno udejstvovanje. Teorije in predpostavke so lažje dokazljive in uporabljeni podatki lažje preverljivi. Z razvojem in množično uporabo tehnologije zelo narašča količina podatkov, ki nastanejo in jih hranimo v digitalni obliki. Ti podatki predstavljajo osnovo mnogih delovnih procesov, zato je pametno poskrbeti, da so čim več časa razpoložljivi. Glede na dejstvo, da do določenih podatkov dostopamo pogosteje kot do drugih, se postavlja vprašanje, ali je smiselno vse podatke hraniti na medijih s hitrim dostopom. Tehnologija nam dandanes ponuja veliko rešitev za obvladovanje vse večje količine podatkov. Znotraj ustanov je zaradi virtualizacijskih tehnologij čedalje večkrat zaznati centralizacijo shranjevanja podatkov, saj nam že sam koncept centralizacije predstavlja večji izkoristek prostora in s tem zniževanje TCO. Z uporabo tehnologij za zmanjševanje količine zapisanih podatkov (kompresija, deduplikacija) in razvrščanje podatkov glede na pogostost uporabe lahko te stroške še dodatno zmanjšamo.

3.1 Razlaga uporabljenih terminov

V tem poglavju razložimo izraze, ki jih bomo uporabljali v nadaljevanju in so ključni za razumevanje tematike.

RTO in RPO

Kritični dogodek (angl. *Disaster*) je definiran kot točka v času, ko je neka storitev nepredvideno prekinjena, onemogočena ali kompromitirana. Pojem RTO pomeni čas, potreben za ponovno vzpostavitev storitve po kritičnem dogodku, pojem RPO pa nam pove, kolikšno količino podatkov/transakcij smo v primeru kritičnega dogodka še pripravljeni izgubiti, oziroma z drugimi besedami, kolikšno izgubo podatkov poslovni proces prenese z razumnimi stroški. Vizualizacija pojmov je predstavljena na sliki 3.1. Z zmanjševanjem RPO in RTO stroški zelo hitro naraščajo.



Slika 3.1 Grafični prikaz RTO in RPO.

Okrevanje po kritičnem dogodku

Okrevanje po kritičnem dogodku (angl. *Disaster Recovery Procedure*) je postopek namenjen delovanju informacijskega sistema, s katerim po kritičnem dogodku povrnemo podatke najmanj v točko RPO in najkasneje v času RTO vzpostavimo delovanje storitve na isti ali drugi lokaciji in opremi. Če je potrebno, definira tudi postopke za vrnitev na primarno lokacijo. Postopek se skladno s prioritetno listo izdela za vse storitve, ki jih informacijska služba zagotavlja poslovnemu procesu.

Načrt neprekinjenega poslovanja

Načrt neprekinjenega poslovanja (angl. *Business Continuity Plan*) je dokument, ki vsebuje informacije o poteku vzpostavitve poslovanja po kritičnem dogodku. Izdelava načrta vsebuje pet korakov:

- analiza vpliva na poslovanje (angl. *Business Impact Analisys*),
- analiza in riziko groženj (angl. *Threat and Risk Analisys*),
- identifikacija ključnih poslovnih funkcij, izdelava prioritetne liste in tabelo odvisnosti,
- določitev lokacije, opreme in osebja za nadaljevanje poslovanja,
- periodično testiranje in osveževanje postopka.

Med analizo vpliva na poslovanje pregledamo finančne posledice kritičnih dogodkov in določimo sprejemljiva RPO in RTO glede na finančne zmožnosti, posledice ter zakonska določila.

Pri analizi groženj se osredotočimo na pričakovane dogodke (požar, potres, smrt ključnih ljudi, epidemije, naklepna dejanja, itd), ocenimo verjetnost, da do njih pride in oblikujemo postopke za obvladovanje situacij.

Prioritetna lista ključnih poslovnih funkcij in tabela odvisnosti sta pomembni zato, da lahko pripravimo ustrezen postopek za okrevanje po kritičnem dogodku in minimiziramo RPO in RTO v skladu z zmožnostmi.

Če je potrebno poslovanje urediti v kratkem času po kritičnem dogodku in v analizi groženj ugotovimo, da je velika verjetnost za dogodek, po katerem vzpostavitev sprejemljivega stanja traja predolgo (npr. porušitev zgradbe, poplava, itd.), se določi nadomestna lokacija za izvajanje poslovnega procesa. Lokacija mora biti ustrezna glede umestitve v prostoru in zmožnosti vzpostavitve delovnih procesov. Sem štejemo prostor, ki je na razpolago za delavce in stranke, ustrezno povezavo z internetom, dostopnost, možnost parkiranja, dovolj zmogljiv električni priključek, možnost postavitve delovne opreme, itd.

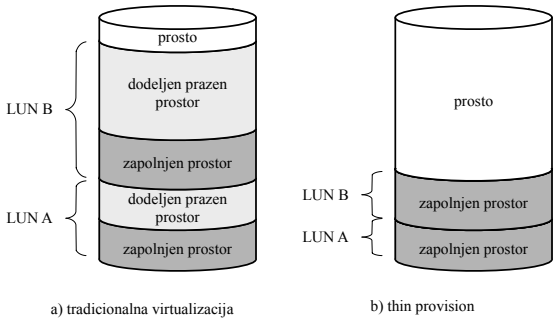
Redno testiranje postopka zagotavlja,

- da so informacije v načrtu pravilne,
- da se vodstvo zaveda vpliva na poslovanje,
- da bodo ključni ljudje pravilno odreagirali,
- da so tehnične rešitve testirane in preverjene,
- testiranje in preverjanje okrevalnih postopkov.

"Thin provisioning"

"Thin provisioning" je način za optimizacijo zasedenosti diskovnih sistemov. Tradicio-

nalna virtualizacija diskovnih sistemov omogoča, da diskovni prostor razdelimo na več virtualnih diskov (LUN) in jih dodelimo različnim odjemalcem. Tak LUN zaseda toliko prostora, kot je definirano z njegovo velikostjo. Običajno se uporabnik preceni v količini podatkov in je zato precej prostora neizkoriščenega, "thin provisioning" pa nam omogoča, da uporabnik vidi željeni prostor, na diskovnem sistemu pa kljub temu zaseda samo toliko prostora, kolikor ima podatkov, kot je prikazano na sliki 3.2. Dokler so zahteve uporabnikov po prostoru manjše od dejanske kapacitete diskovnih sistemov, je ta tehnika brezpredmetna, v veljavo pride, ko imamo uporabnike z velikimi zahte-

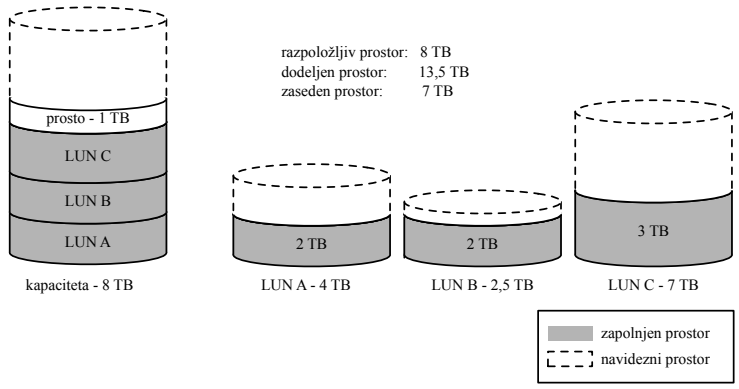


Slika 3.2 Primerjava zasedenosti diskovnega polja med tradicionalno in "thin provisioned" virtualizacijo.

vami, a bolj malo podatki. Tradicionalni sistem bi bil v tem primeru zelo neizkoriščen, "thin provisioning" pa nam omogoča, da z manjšo kapaciteto zadovoljimo uporabnike in kljub temu prihranimo pri nakupu stojne opreme. Paziti moramo le, da dovolj hitro sprožimo postopek za nakup diskov, ko se porabljen prostor približuje dejanski kapaciteti diskovnega sistema. Pojav, ko je vsota velikosti definiranih LUN-ov večja od dejanske kapacitete sistema, angleško imenujemo "over provision". Primer je na sliki 3.3.

Deduplikacija

Deduplikacijo lahko opišemo kot posebno tehniko kompresije, s katero iščemo ponavljajoče se vzorce. Izvajamo jo lahko na datotekah ali na blokih, zapisanih na diskih. Deduplikacija na blokih je učinkovitejša, saj zajame tudi enake dele zapisanih datotek. Bloki so obdelani z "hash" algoritmom, ki ustvari unikaten ključ za vsak blok in ga zapiše v tabelo vzorcev. Vsak blok se pred zapisom preveri v tabeli vzorcev in če najde ujemajoč se podatek, zapiše samo kazalec. S to tehniko pridobimo na izkoristku prostora, najbolj se to pozna pri zagonskih diskih navideznih strežnikov in pri varnostnih kopijah



Slika 3.3 "Over provisioning".

podatkov, kjer je veliko kopij istega oz. zelo podobnega podatka.

Vpliv latence

Latenca je čas, ki preteče od oddaje zahteve do prejema odziva. Vpliv latence na dostop do podatkov se nam zdi majhen, vendar lahko pri večjih količinah podatkov ali pri zahtevah po hitrem dostopu precej vpliva na zmogljivosti. Dolge povezave običajno potekajo po optičnih vlaknih, zato se bomo omejili na to tehnologijo. Na latenco optičnih povezav vplivajo

- hitrost potovanja signala po mediju,
- optične komponente na povezavi,
- naprave za pretvorbo signala iz svetlobe v elektriko in obratno,
- algoritmi za odpravo napak [7].

Signal po optičnem vlaknu potuje s hitrostjo okrog 66% svetlobne hitrosti. Hitrost je definirana z lomnim količnikom, ki je dovolj natančno definiran z vrednostjo 1,47 in znaša približno

$$v = \frac{c}{n} = \frac{299792458}{1,47} = 203940667\frac{m}{s} \tag{3.1}$$

oz. 4,9 ns/m, kar lahko zaokrožimo na 5 ns/m.

Pri vsakem prenosu svetlobe prihaja do disperzije, ki z razdaljo narašča. Na daljših razdaljah je zato potrebno poskrbeti za zmanjševanje disperzije svetlobe. Za ta namen se uporabljajo t.i. filtri za kompenzacijo disperzije (angl. *Dispersion Compensating Filter*),

ki pa povečajo latenco za 20-25%. Z razvojem tehnologije je bil razvit modul za kompenzacijo disperzije (angl. *Dispersion Compensation Module*), ki ravno tako učinkovito opravi disperzijo, le da to naredi brez dodatne latence.

Druga težava pri dolgih razdaljah je ojačevanje signala. Če ga ojačujemo s pretvorbo v elektriko in nazaj, to prinese dodatno latenco nekaj μs, zato raje uporabimo ojačevalce, ki ne vplivajo na latenco (Raman ojačevalci).

Pretvorba svetlobe v elektriko se na poti signala zgodi vsaj dvakrat v vsako smer - v vstopu v optično povezavo in pri izstopu iz nje, če uporabljamo ojačevalce s pretvorbo v elektriko, pa še dvakrat na vsakem ojačevalcu. Vstop je lahko že na adapterju ali pa kasneje na poti signala.

Pri daljših linijah (nekaj 100 km) je potrebno uporabljati FEC (angl. *Forward Error Correction*), ki prinese latenco do 100 μs.

Vzemimo primer zapisovanja podatka na dva diska (zrcaljenje). Prvi disk se nahaja v istem prostoru, drugi pa na 100 km oddaljeni lokaciji. Razdalja med strežnikom in prvim diskom naj bo 10 metrov. Čas potovanja signala (od zahteve do potrditve) do prvega diska je 100ns, do drugega pa 1 ms ob predpostavki, da imamo idealno linijo. Če zapišemo blok velikosti 64KB in imamo povezavo med lokacijama 1 Gbps, na posamezni lokaciji pa 8Gbps, dobimo čas, ki ga blok potrebuje, da prepotuje linijo:

$$t = cas\ potovanja\ prvega\ bita + \frac{kolicina}{prepustnost}. \tag{3.2}$$

Čas za prenos podatka v lokalnem okolju je

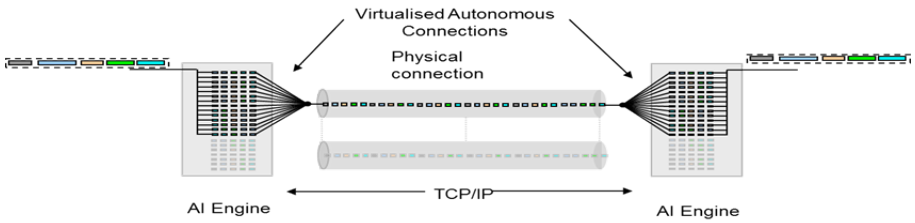
$$t_1 = 0,0001 + \frac{524228}{8 \cdot 10^9} = 0,0001655285s \simeq 166\mu s. \tag{3.3}$$

Čas za prenos podatka na oddaljeno lokacijo je

$$t_2 = 0,001 + \frac{524228}{10^9} = 0,001524288s \simeq 1524\mu s. \tag{3.4}$$

Zmanjševanje vpliva latence

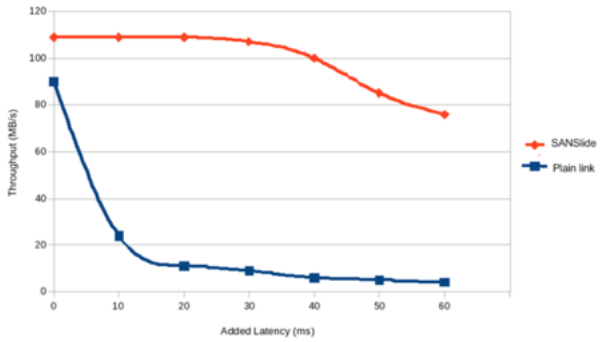
Daljša kot je povezava, večja je latenca in kasneje pride potrditev, da je bilo pošiljanje paketa uspešno. Na SAN povezavah, kjer je v uporabi FC protokol, se ta težava rešuje z uporabo "Buffer-to-Buffer Credit". To je strojna metoda kontrole toka, ki pošiljatelju omogoča, da pošlje določeno število paketov (angl. *Buffer Credit*), preden dobi potrditev od prejemnika. Ko pošiljatelj pošlje paket, zmanjša "Buffer Credit" za ena, ko prejme



Slika 3.4 Prikaz izkoriščanja linije s SANSlide [8].

potrditev, ga poveča. Pakete lahko pošilja, če ima kakšen "Buffer Credit" na voljo, sicer čaka na potrditev. Število je odvisno od deklarirane zmogljivosti in dolžine linije.

Pri TCP/IP povezavi te možnosti ni. Prepustnost dolge linije je možno povečati z uporabo namenske strojne opreme, IBM pa je uporabil programsko rešitev SANrockIT[1]. Slika 3.4 prikazuje tehnologijo - ustvari se več hkratnih navideznih IP povezav in znotraj vsake povezave se čaka na potrditev. Kot je prikazano na sliki 3.5, se prepustnost linije z veliko latenco občutno dvigne.



Slika 3.5 Graf prepustnosti SANRockIT [8].

"Copy-on-write"

"Copy-on-write" je tehnologija zapisovanja podatkov, ki omogoča souporabo delov datotek različnim lastnikom. Dokler več procesov bere isto datoteko, ni težav. Vsakemu procesu se ustvari kazalec, ki kaže na vsebino. Ko želi proces shraniti del podatka, se ustvari privatna kopija vsebine. Ob zaprtju datoteke se spremenjen podatek shrani ločeno, nespremenjen del pa je shranjen samo na izvoru. S tem se izognemo podvajanju podatkov in prihranimo prostor [9].

[1] Pred preimenovanjem se je imenovala SANslide

Temperatura podatkov

Za potrebe razvrščanja podatkov je uveden pojem *temperatura podatkov*. Po temperaturi razdelimo podatke v štiri skupine:

- vroči podatki (angl. *hot data*),
- topli podatki (angl. *warm data*),
- mrzli podatki (angl. *cold data*) in
- mirujoči podatki (angl. *dormant data*).

Tabela 3.1 prikazuje značilnosti posamezne skupine podatkov.

vrsta	***značilnost***	***običajna starost***
vroči	sveži podatki za takojšnje in kratkoročne obdelave	do 3 mesece
topli	namenjeni za dolgoročne odločitve ali raziskave, obdelave ne rabijo hitrega odziva	3 mesece do 1 leto
mrzli	zgodovinski podatki	1 do 5 let
mirujoči	arhivski podatki ali podatki, ki jih moramo hraniti po zakonu	nad 5 let

Tabela 3.1 Značilnosti skupin podatkov [10].

Temperatura sovpada s pogostostjo uporabe podatka. Povprečna temperatura ima tendenco padanja, ker se podatek seli proti mirujočim podatkov. Podatke lahko med različnimi skupinami selimo ročno ali pa jih selijo procesi, ki nadzorujejo njihovo temperaturo.

Distribuirana RAID polja

RAID polja omogočajo varnost podatkov v primeru odpovedi diska. Odvisno od vrste RAID polja lahko odpove različno število diskov, preden pride do izgube podatkov. Čas med odpovedjo diska in zamenjavo le-tega z delujočim je iz vidika varovanja podatkov zelo kritičen. Diskovna polja omogočajo določitev t.i. rezervnih diskov (angl. *hot spare*) - to je v disk v diskovnem polju, ki čaka, da bo nadomestil morebitni okvarjen disk. Ob odpovedi se vsebina okvarjenega diska prepiše (oz. izračuna) na “hot spare”. V tradicionalnih RAID poljih to pomeni, da se podatki berejo iz vseh diskov in zapisujejo

na en disk. Med okrevanjem so zmogljivosti do uporabnikov okrnjene, čas za okrevanje pa je odvisen od velikosti diska.

Diskovni sistemi s tradicionalnimi RAID polji in z velikimi diski imajo dve slabosti:

- čas okrevanja diska,
- rezervni diski niso izkoriščeni.

Diski zmorejo sekvenčne podatke zapisovati s hitrostjo okrog 120MB/s. Teoretičen čas za okrevanje 300GB diska je

$$\frac{300GB}{120\frac{MB}{s}} = 2560s \simeq 42minut, \tag{3.5}$$

za 8 TB disk pa

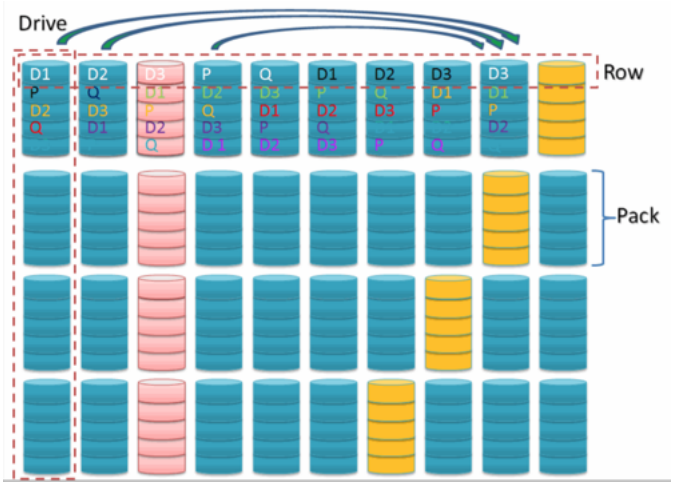
$$\frac{8TB}{120\frac{MB}{s}} = 69905s \simeq 19,5ur. \tag{3.6}$$

V distribuiranem RAID polju so podatki in rezervni prostor razdeljeni med vse diske v polju. Ideja je, da RAID polje dolžine X naredimo na Y diskih, pri čemer je Y večji od X. Primer RAID6 dolžine 5 razpršen na 10 diskov je prikazan v sliki 3.6.

Slika 3.6 Prikaz distribuiranega RAID polja [11].

V primeru odpovedi diska se manjkajoči del podatka izračuna iz preostalih delov in zapiše v rezervni prostor na drugem disku. Zaradi večjega števila diskov, kot je dolžina RAID polja, je na odpovedanem disku samo del podatkov, kar pomeni, da je potrebno manj računati.

S tem načinom odpravimo težave tradicionalnih RAID polj:



Slika 3.7 Prikaz postopka ob odpovedi diska v distribuiranem RAID6 polju [11].

- rezervni diski so vključeni v polje kot rezervni prostor, zato je na voljo več diskov za izvajanje vhodno/izhodnih operacij,
- v primeru odpovedi se podatki iz več diskov kopirajo na več diskov,
- čas okrevanja diska se zmanjša (do desetkrat [11]),
- vpliv na zmogljivosti je zmanjšan.

3.2 Možnosti za povečanje razpoložljivosti podatkov

V regijskih bolnišnicah je velik pretok pacientov. Paciente mnogokrat obravnavajo v različnih ambulantah, zato je nujno, da so interoperabilni podatki med obravnavo razpoložljivi. Razpoložljivost je definirana kot stanje, v katerem so uporabniki zmožni dostopati do opazovanega sistema. Običajno se podaja v odstotkih časa, predstavlja pa verjetnost za neplanirano nedostopnost sistema v časovnem obdobju. Če želimo zagotoviti 99,99% razpoložljivost, to pomeni, da si lahko v dolgoročnem povprečju privoščimo izpade

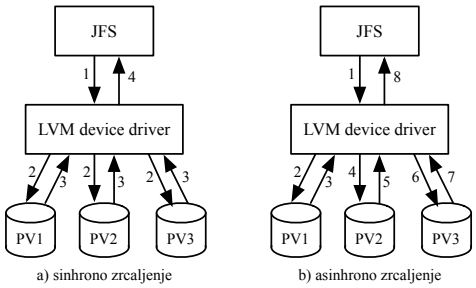
- dnevno: 8,6s,
- tedensko: 1m 0,5s,
- mesečno: 4m 23s,
- letno: 52m 35,7s.[2]

[2] http://uptime.is/99.99

Pojem razpoložljivosti bi se utegnil mešati s pojmom "uptime" (čas, v katerem je določen sistem operativen), vendar na samo razpoložljivost neke storitve vpliva več dejavnikov, npr. izpad omrežja, izpad strežnika, ki to storitev ponuja, izpad podpornih storitev, itd. Pojma "uptime" in razpoložljivost (angl. *availability*) torej med seboj nista nujno povezana. Razpoložljivost podatkov je možno povečevati na več načinov, odvisno od tega, kaj zahteva načrt neprekinjenega poslovanja.

3.2.1 Replikacija podatkov na nivoju operacijskega sistema

Zrcaljenje podatkov je najenostavnejši način za zagotavljanje razpoložljivosti podatkov v primeru izpada enega izmed podatkovnih nosilcev. Zrcaljenje lahko poteka sinhrono ali asinhrono. Sinhrono zrcaljenje nam zagotavlja RTO 0 in RPO brez izgube podatkov. Z asinhronim zrcaljenem se tema dvema vrednostima lahko približamo, izenačimo pa samo v primeru, da imamo dovolj zmogljivo povezavo, ki v dovolj kratkem času prenese vse spremembe na primarnem mediju na sekundarni medij. Razlika je prikazana na sliki 3.8. Možne so tudi kombinirane rešitve.

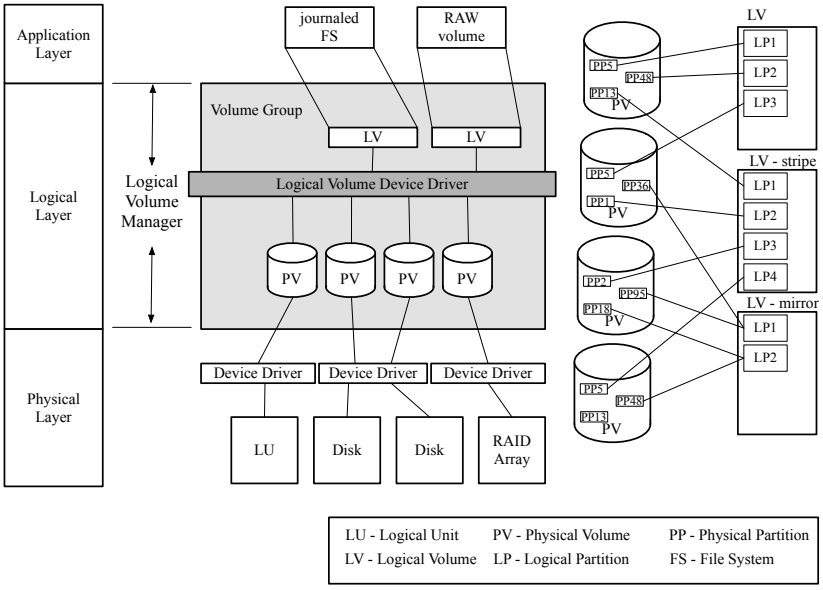


Slika 3.8 Razlika med sinhronim in asinhronim zrcaljenjem [12].

Operacijski sistemi z uporabo aplikacijskih rešitev ponujajo tudi druge možnosti implementacije RAID polj (4, 5, 6, 10). Te implementacije ponujajo boljše zmogljivsti in večjo verjetnost za napako kot zrcaljenje (RAID 1), saj je vanje vključeno več komponent. Skupna verjetnost za odpoved je namreč produkt verjetnosti za odpoved posameznih komponent. Vsem rešitvam na nivoju operacijskega sistema je skupno, da so razdrobljene in jih ni možno upravljati iz enega mesta, zato so primerne za manjše uporabnike.

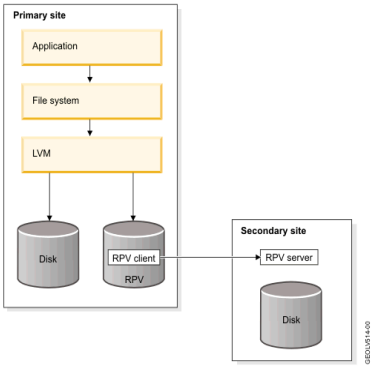
Replikacija podatkov z uporabo LVM

Operacijski sistemi za to opravilo uporabljajo eno od različic LVM. To je del programske kode, ki omogoča združevanje več diskov ("Physical volume") v eno entiteto ("Volume Group"), delitev te entitete na več logičnih enot ("Logical Volume"), implementacijo različnih RAID struktur na nivoju "Logical Volume" in upravljanje z datotečnimi sistemi ("File System") na "Logical Volume". Prikaz LVM je na sliki 3.9. Enostavnešje implementacije (npr. mdraid) omogočajo samo delo s celimi diski. LVM je lahko sestavni del operacijskega sistema, lahko pa je tudi samostojen komercialni produkt.



Slika 3.9 Shematski prikaz LVM [13].

Bistvena prednost LVM je ta, da nam omogoča spreminjanje velikosti datotečnega sistema med delovanjem in s tem neposredno vpiva na razpoložljivost podatkov. Na UNIX strežnikih se to zelo pogosto uporablja, ker omogoča zelo enostavno delitev datotečnih sistemov po namenu uporabe, s tem pa preprečimo, da bi en program onemogočil izvajanje drugih programov z zapolnitvijo datotečnega sistema. Z uporabo LVM povečamo razpoložljivost podatkov na lokalnih diskih ali diskovnih poljih, ki med seboj niso interoperabilna. Zagotavljanje, da se zrcaljeni podatki nahajajo na različnih diskih oz. diskovnih poljih (in s tem na različnih lokacijah) je v primeru večih diskov zahtevno. Posebej ob urgentnih povečevanjih datotečnih sistemov je to precej stresno in časovno



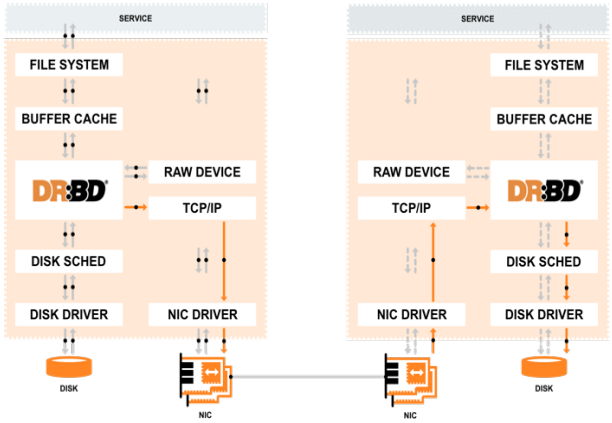
Slika 3.10 Shematski prikaz GLVM [14].

potratno. V primeru sinhronega zrcaljenja med dvema lokacijama je potrebno upoštevati še performančne izgube, sploh če repliciramo na daljše razdalje, kjer se precej poveča latenca pri zapisu podatkov. Replikacija na nivoju LVM poteka po diskovnih protokolih in zato zahteva namensko linijo in ustrezno opremo. IBM je razvil možnost za uporabo TCP linije med lokacijama. Vpeljal je pojem oddaljenega diska ("Remote Physical Volume") in ga integriral v obstoječ LVM. Na sliki 3.10 je predstavljen koncept, ki ga je poimenoval GLVM - Geographic LVM.

Replikacija podatkov s prestrezanjem sprememb v datotekah

Te rešitve se poslužujejo principa vmesnega člena med datotečnim sistemom in gonilnikom za disk, ki poskrbi, da gre zahteva za zapis podatka preko omrežne povezave (IP, Infiniband, ATM, ipd.) še na disk na oddaljeni lokaciji. Primer je odprtokodna rešitev podjetja Linbit [15]: programska koda prestreže podatek na poti med datotečnim sistemom in diskom in ga po TCP protokolu pošlje na drugo lokacijo (slika 3.11). Namesto klasičnega TCP protokola so na voljo tudi protokoli, optimizirani za prenose na daljše razdalje, kot so Aspera® FASP® in FASP3^{TM}, uporabljajo UDP del 4. nivoja OSI in z lastnimi rešitvami poskrbijo za zagotavljanje dostave paketov [16]. S paralelizmom se izognejo latenčnim težavam in tudi dolgo linijo zasedejo skoraj v celoti. Pakete kompresirajo, na njih izvajajo deduplikacijo in jih zaščitijo z uporabo kriptirnih algoritmov [17]. Težave se pokažejo pri replikaciji odprtih datotek, še posebej pri bazah z več hkrati odprtimi datotekami, ki jih je praktično nemogoče vzdrževati v konsistentnem stanju, zato so te rešitve namenjene za uporabo z datotečnimi strežniki.

Teoretično je razdalja med obema diskoma neomejena, praktično pa smo omejeni s



Slika 3.11 Shematski prikaz DRBD [15].

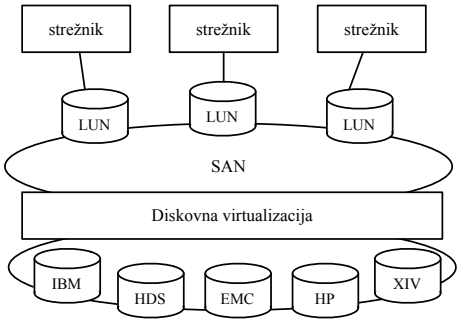
pričakovanim odzivnim časom pri sinhronem in dovoljenim RPO pri asinhronem zapisovanju. V kontroliranem postopku nimamo izgube podatkov, RTO pa je odvisen od postopka prehoda na rezervno lokacijo.

Podvajanje datotek z aplikacijami

Podvajanje datotek z aplikacijami je semantično najenostavnejša rešitev. Težava nastopi, ko je potrebno replicirati veliko število datotek, velike ali odprte datoteke. Tudi pogosta replikacija ali uporaba po visokolatenčnih linij sta lahko vir težav. Uveljavljene metode, kot je rsync, pokažejo slabosti, ko je potrebno nadzorovati veliko datotek. Takrat lahko že preprosto preimenovanje direktorija povzroči daljše zamude pri repliciranju, ker je potrebno ponovno prebrati in primerjati vsebino direktorijev.

3.2.2 Replikacija podatkov na nivoju diskovnega sistema

Razvoj procesorjev in komunikacijskih tehnologij je prinesel veliko možnosti za prilagajanje in razvoj dodatnih zmožnosti tudi na nivoju shranjevanja podatkov, kjer nismo več omejeni na strojno implementacijo komunikacij in zaščite zapisanih podatkov, ampak nam programska oprema omogoča prenos čedalje več funkcij na diskovne sisteme. Nivo virtualizacije se je razširil na virtualizacijo med diskovnimi sistemi (slika 3.12). To nam omogoča enostavo upravljanje s podatki in replikacijo podatkov praktično kamorkoli. Omejeni smo samo z zmogljivostjo povezave. Tovrstne rešitve imajo danes praktično vsi resni proizvajalci diskovnih sistemov, v nalogi pa se bomo omejili na rešitve podjetja IBM.



Slika 3.12 Virtualizacija na nivoju diskovnih sistemov [18].

IBM s programsko opremo Spectrum Virtualize omogoča naslednje načine podvojevanja podatkov [19]:

- FlashCopy,
- Metro Mirror,
- Global Mirror,
- Global Mirror with Change Volumes,
- Stretched Cluster,
- Hyperswap.

FlashCopy

FlashCopy je zmožnost kreiranja presekov stanja na diskih v zelo kratkem času. Iz izvornega LUN-a skopira podatke na ciljni LUN v določeni točki v času, medtem ko so podatki na izvornem LUN-u na voljo za uporabo. Na voljo so trije načini izvajanja operacije: trajni prepis ali klon (angl. *clone*, tudi Copy), začasni posnetek (angl. *snapshot*, tudi noCopy) in stalna varnostna kopija (angl. *continuous backup*). Klon in stalna varnostna kopija zahtevata, da sta izvorni (angl. *source*) in ciljni (angl. *target*) LUN enako velika, začasni posnetek pa dovoljuje, da uporabimo manjši ciljni LUN. Zahtevana velikost je odvisna od intenzivnosti pisanja v času, ko obstaja začasni posnetek. Za krajša časovna obdobja je priročno ustvariti "thin provisioned" LUN.

Za zagotavljanje konsistence podatkov na ciljnem LUN-u FlashCopy zamrzne dostop do izvornega LUN-a, naredi bitmap tabelo v velikosti števila blokov na izvornem LUN-u in vanjo vpiše vrednosti 0 [20]. V tem trenutku je ciljni LUN pripravljen za uporabo, izvorni LUN pa je spet normalno dostopen. Na ciljni LUN se podatki zapišejo, preden

pride do spremembe na izvornem LUN-u, pri kloniranju pa se v vmesnem času kopirajo še ostali podatki. Ob prepisanem podatku se spremeni vrednost v bitmap tabeli. Pomen vrednosti se razlikuje glede na način izvajanja FlashCopy operacije. Razlaga je v tabeli 3.2.

način izvajanja	***vrednost***	***pomen***
klon (Copy)	0	Podatek ni spremenjen in še ni skopiran, branje podatka iz izvornega LUN-a.
	1	Podatek je že skopiran ali pa je bil spremenjen na izvornem LUN-u. Branje iz ciljnega LUN-a.
začasni posnetek (noCopy)	0	Podatek ni spremenjen, branje podatka iz izvornega LUN-a.
	1	Podatek je bil spremenjen na izvornem LUN-u, stari podatek je skopiran, branje iz ciljnega LUN-a.

Tabela 3.2 Pomen vrednosti v bitmap tabeli.

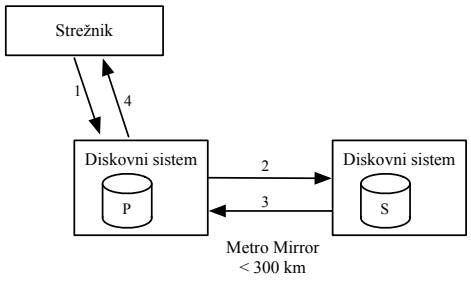
Če so izvorni podatki zapisani v več LUN-ih, se vse LUN-e združi v konsistenčno grupo, ki nam zagotavlja, da se bo FlashCopy izvedel na vseh LUN-ih istočasno. FlashCopy se lahko izvaja v obe smeri.

Uporabna vrednost FlashCopy je v zagotavljanju hitre povrnite podatkov v primeru napake med planiranim postopkom, izdelavi varnostnih kopij na drugem strežniku, izvajanju dolgih obdelav ob zagotavljanju konstantnih podatkov, kloniranju podatkov med delovanjem, itd.

FlashCopy je uporabna tehnika za varovanje podatkov med planiranimi deli, ker nam zagotavlja RPO v zadnje stanje in RTO nekaj sekund. Za varovanje pred neplaniranimi dogodki je manj primeren.

Metro Mirror

Metro Mirror je način sinhronega podvojevanja podatkov med primarnim in sekundarnim LUN-om. Sekundarni LUN je lahko na istem ali drugem diskovnem sistemu. Pri tem načinu podvojevanja podatkov odjemalec, ki se ne zaveda, da ima podatke podvojene, pošlje zahtevo po zapisu podatka na primarni LUN, potrditev pa dobi šele, ko je podatek zapisan na sekundarni LUN, kot je prikazano na sliki 3.13. Ta način zagotavlja, da so podatki vedno sinhronizirani. Branje vedno poteka iz primarnega LUN-a.



Slika 3.13 Potek zapisa podatka z uporabo Metro Mirror [21].

Metro Mirror je omejen z razdaljo 300km. Zaradi sinhronega zapisovanja je podobno kot pri zrcaljenju z LVM potrebno računati na vpliv na performance, tako da je uporaba tega načina običajno omejena na krajše razdalje. Zahteva visokoprepustne linije z nizko latenco.

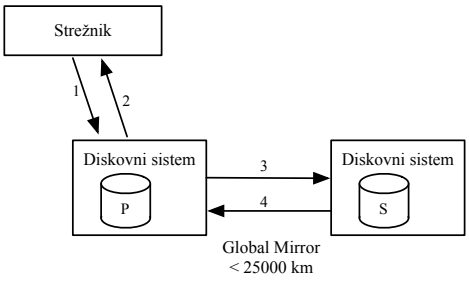
RPO je, tako kot pri zrcaljenju z LVM, brez izgube, RTO pa je odvisen od postopka za prehod na rezervno lokacijo in je okvirno med minutami in urami.

Global Mirror

Kadar je vpliv na performance sistema pri sinhronem podvojevanju podatkov prevelik, lahko uporabimo Global Mirror. Tukaj se podatki med primarnim in sekundarnim LUN-om kopirajo asinhrono - odjemalec takoj dobi potrditev, da je podatek zapisan, diskovni sistem pa naknadno poskrbi, da se podatek zapiše še na sekundarno kopijo in posodobi tabelo sinhroniziranih podatkov na obeh straneh (slika 3.14). Količina zahtev po pisanju, do katere je Global mirror sposoben vzdrževati sinhronizirano kopijo podatkov, je odvisna od zmogljivosti SAN povezave med diskovnima sistemoma. Določa jo najdaljši čas, v katerem je dopustna vnaprej določena zakasnitev pri pisanju na primarni LUN. Privzeto 5 minut dopušča 5 ms zakasnitev [22]. Po preteku omejitve se Global Mirror relacija za ta LUN ustavi. RPO je običajno v razredu milisekund, je pa odvisen od intenzivnosti pisanja. Zahteva visokoprepustne linije, latenca pa je lahko višja kot pri Metro Mirrorju. RTO je tako kot pri Metro Mirrorju odvisen od postopka prehoda na rezervno lokacijo.

Global Mirror with Change Volumes

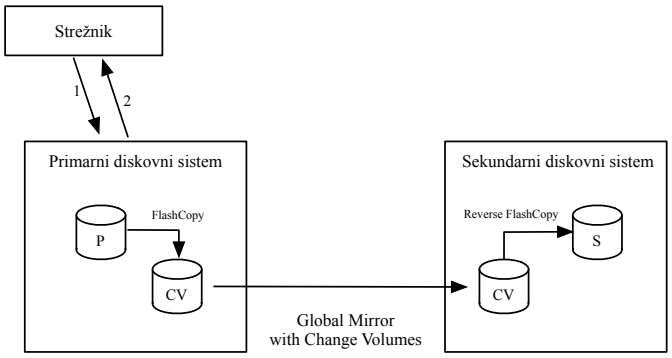
Kadar prepustnost linije med primarno in rezervo lokacijo ne ustreza našim zahtevam (finančno ali časovno), si lahko pomagamo z Global Mirror with Change Volumes (GMCV).



Slika 3.14 Potek zapisa podatka z uporabo Global Mirror [21].

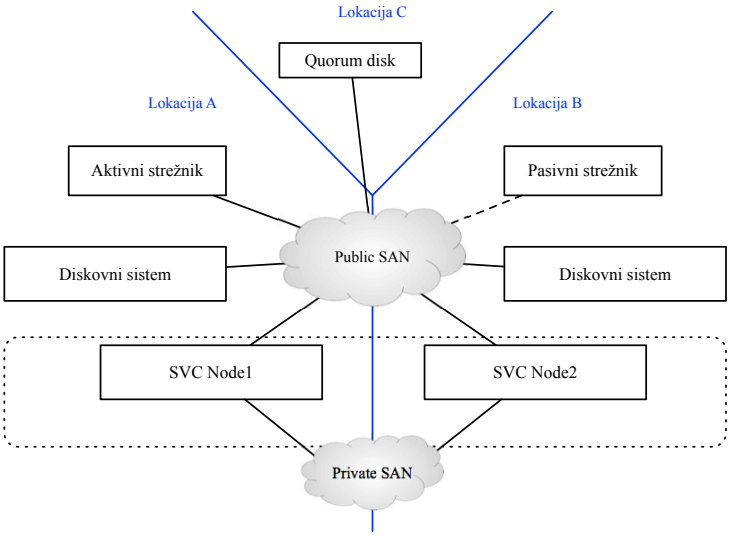
Ta način je podoben Global Mirrorju, vendar z razliko, da kopiranje podatkov ni konstantno, ampak periodično. Perioda je nastavljiva na 10 sekund v razponu od 1 minute do 24 ur. Princip podvajanja podatkov je predstavljen na sliki 3.15. Ob aktiviranju relacije se naredi FlashCopy primarnega LUN-a na t.i. Change Volume (CV). To je "thin provisioned" LUN, ki služi za zagotavljanje konsistence podatkov med podvojevanjem na sekundarni CV LUN. Ko so podatki prenešeni, se na sekundarni strani izvede Reverse FlashCopy - postopek, ko podatke iz FlashCopy LUN-a prepišemo čez podatke na izvornem LUN-u.

Med do sedaj obravnavanimi načini podvojevanja podatkov je ta najmanj zahteven glede linije. To je tudi edini način, kjer lahko podatke podvajamo po TCP protokolu.



Slika 3.15 Potek zapisa podatka z uporabo Global Mirror [23].

RTO je pri uporabi GMCV enak kot pri Metro- ali Global Mirrorju, RPO pa je



Slika 3.16 Prikaz arhitekture Stretched Cluster (prim. [22]).

določen z izrazom:

$$RPO = \begin{cases} perioda \leq RPO \leq 2 \cdot perioda, & kopiranje \leq perioda \\ vsota\ dveh\ kopiranj, & sicer \end{cases} \tag{3.7}$$

"Stretched Cluster"

Praktično vsi diskovni sistemi imajo podvojene komponente, da zadostijo osnovnim zahtevam po visoki razpoložljivosti in vzdrževanju med delovanjem. Ideja koncepta "Stretched Cluster" [22] je, da razdvojimo oba kontrolerja, vsakemu od njiju dodelimo svoj diskovni sistem in med njima zrcalimo vsebino. Shematski prikaz je na sliki 3.16. Ker sta kontrolerja lahko na različnih lokacijah (dovoljena razdalja med obema kontrolerjema je določena z načinom povezave in znaša 40 ali 300km) je za preprečevanje t.i. "split brain" konfiguracije obvezna uporaba "quorum" diska na tretji lokaciji. Za "Stretched Cluster" lahko izberemo samo IBM San Volume Controler (SVC), ostali diskovni sistemi imajo oba kontrolerja vgrajena v istem ohišju.

Tak način podvajanja podatkov ima veliko prednosti. Tako kot zrcaljenje z LVM v primeru izgube ene kopije ponuja brezizgubni RPO in zelo kratek RTO. Podvajanje podatkov deluje v obe smeri. Vsebino posameznih LUN-ov lahko kljub temu, da so že podvojeni, z uporabo Metro- ali Global Mirrorja repliciramo na tretjo lokacijo.

Med slabostmi poleg zahteve po zelo zmogljivih linijah med obema kontrolerjema iz-

stopa ta, da LUN-ov ne moremo združevati v konsistenčne grupe, ker niso v t.i. "remote mirror" relaciji. V primeru, da eden izmed LUN-ov, dodeljenih isti aplikaciji, zaradi intenzivnega pisanja ne more vzdrževati sinhronega stanja, tega ostali LUN-i ne detektirajo in posledično so podatki določen čas nekonsistentni. Druga slabost je, da v primeru izgube enega kontrolerja vse zrcaljene LUN-e sinhronizira od začetka, tretja pa, da v primeru izgube enega kontrolerja onemogoči pisalni predpomnilnik, kar vodi v zmanjšanje zmogljivosti.

Hyperswap

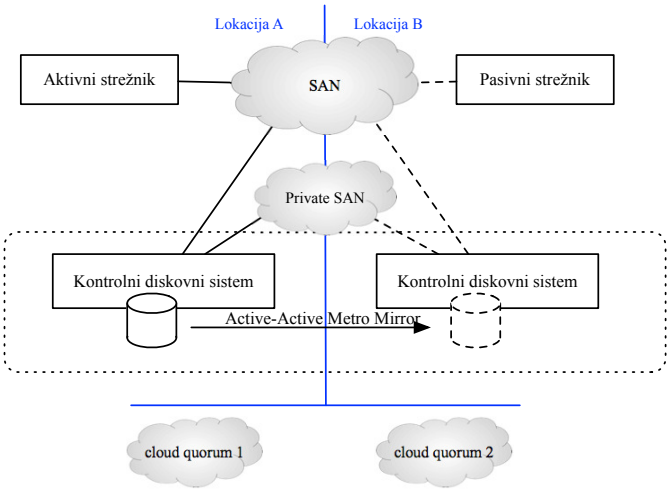
Hyperswap je koncept, ki ga IBM že precej časa uporablja na mainframe okoljih, kjer je pričakovana razpoložljivost zelo blizu 100%. Tukaj gre dejansko za kombinacijo vseh zgoraj naštetih tehnik za podvajanje podatkov. Uporablja dva diskovna sistema, ki sta med seboj povezana v gručo. Ravno tako kot Stretched system zahteva "quorum" disk, ki pa je lahko tudi prostor v oblaku[3] (slika 3.17). S tem odpade potreba po tretji lokaciji. Za podvajanje vsebine LUN-a uporablja Metro Mirror s hkratno uporabo "Change Volume". To prinaša dve prednosti in eno slabost:

- LUN-e lahko združujemo v konsistenčne grupe (ker dejansko uporablja "remote mirror"),
- v primeru vzpostavitve po izgubi enega diskovnega sistema repliciramo samo razlike in ne celega LUN-a,
- zaradi uporabe "remote mirror" ne moremo podatkov kopirati na tretjo lokacijo.

Uveden je koncept zavedanja strani - to pomeni, da vse komponente vedo, na kateri lokaciji so in iz katere lokacije prihajajo zahteve. Če je le možno, se zahteva servisira lokalno. Hyperswap se nastavlja na nivoju LUN-a in ni potrebe po replikaciji celotnega sistema. Hyperswap volume je v aktivnem načinu na obeh straneh, to pa pomeni, da ob selitvi na rezervno lokacijo za dostop do podatkov ni potrebnen poseg na diskovnem sistemu, ker se Metro Mirror lahko izvaja v obe smeri. Branje in pisanje se izvajata iz lokalne kopije. Če se strežnik preseli na rezervno lokacijo, se po določeni količini prometa Metro Mirror obrne in vhodno-izhodne operacije se preusmerijo na rezervno lokacijo. Vmes so za kratek čas zmogljivosti omejene. Vsak diskovni sistem uporablja svoj predpomnilnik, zato v primeru izpada ene lokacije zmogljivosti niso prizadete.

[3] Podatek iz februarja 2016

RPO je brez izgube podatkov in tudi RTO je lahko blizu 0, če le aplikacija to omogoča.



Slika 3.17 Prikaz arhitekture Hyperswap (prim. [18]).

3.2.3 Replikacija podatkov v podatkovnih bazah

Podatkovne baze vse svoje aktivnosti beležijo v log datoteke, da so v primeru napake zmožne podatke povrniti v konsistentno stanje. Če transakcije, zapisane v teh datotekah, izvajamo na kopiji baze, lahko to kopijo vzdržujemo v takem stanju, kot je izvorna baza. Take baze nam nudijo veliko možnosti uporabe:

- zagotavljajo kratek RTO v primeru okvare izvorne baze,
- zagotavljajo stabilne podatke na dolgoročnih obdelavah,
- omogočajo varovanje pred aplikativnimi napakami,
- omogočajo branje podatkov brez obremenitve izvorne baze,
- razbremenijo izvorno bazo med izvajanjem varnostnega kopiranja podatkov,
- ponujajo realne podatke za testne namene.

"Split mirror"

Ta tehnika nam omogoča, da na nivoju infrastrukture razdelimo zrcaljeni kopiji podatkov. Tehnika je na voljo na nivoju datotečnih sistemov in na diskovnih sistemih. Za zagotovitev konsistence podatkov je potrebno onemogočiti vsa pisanja v trenutku, ko razdvajamo zrcaljeni kopiji. Čeprav je postopek kratek, se prekinitev ne zgodi na vseh

podatkih istočano, zato bi lahko ob omogočenem pisanju prišlo do nekonsistence. Podatkovne baze nam omogočajo, da prekinemo zapis v podatkovne datoteke in s tem zagotovimo konsistenco, medtem ko so izvorni podatki razpoložljivi uporabnikom. Postopek izvajanja tehnike je tak:

- ustavimo vse pisalne operacije na izvorni podatkovni bazi,
- razdelimo zrcaljeni kopiji,
- dovolimo pisalne operacije na izvorni podatkovni bazi,
- inicializiramo kopijo za uporabo.

Ustavljanje pisalnih operacij pomeni, da se v času, ko pisanje v podatkovne datoteke ni dovoljeno, vse pisanje izvaja v začasno pomnilniško strukturo [24]. Tu moramo zagotoviti dovolj prostora, sicer se zgodi, da aplikacije ne morejo zapisati podatka in se ustavijo. Uporabniki to občutijo kot veliko latenco.

Časovna veljavnost kopije podatkov je različna glede na namen uporabe. Pri izvajanju daljših poročil ali varnostnih kopij so podatki nepotrebni takoj po opravljeni nalogi. Za testne namene je kopija podatkovne baze uporabna, dokler ni potrebno osvežiti podatkov. Če želimo kopijo uporabiti kot "standby" bazo, mora imeti dostop do arhivskih log datotek izvorne baze. Z vsebino le-teh jo je potrebno osveževati, da zagotovimo dovolj kratek RPO.

Replikacija informacij o transakcijah

Podatkovne baze beležjo vse transakcije. Te informacije je možno pošiljati "standby" bazi. Ta način podvojevanja podatkov ima določene prednosti pred repliciranjem na nižjih nivojih:

- prenašamo samo spremembe v transakcijskih log datotekah, medtem ko tehnike repliciranja na nižjih nivojih prenašajo isto spremembo še v podatkovnih datotekah in arhivskih log datotekah,
- zagotovimo kontrolo vsebine,
- nismo omejeni na proizvajalca in cenovni razred diskovnega sistema na rezervni lokaciji.

Informacije o transakcijah je možno prenašati na več načinov. Podatke lahko zapisujemo sinhrono ali asinhrono na obe bazi istočasno, lahko prenašamo neaktivne log datoteke

("archive log") in transakcije izvajamo na "standby" bazi, lahko pa neaktivne log datoteke sinhroniziramo z orodji na nižjih nivojih in jih uporabimo pred aktiviranjem "standby" baze. V primeru prenašanja datotek je priporočljivo uporabiti zrcaljenje aktivnih log datotek ("active log", "redo log"), da zagotovimo pravo informacijo o transakcijah tudi v primeru okvare aktivne log datoteke. Informacije o transakcijah lahko na "standby" bazah uveljavimo takoj ali pa z zamikom [25]. Prednost takojšnje uporabe je v zelo kratkem RTO, prednost uporabe z zamikom pa je varovanje podatkov v izvorni bazi pred namerno ali nenamerno okvaro. Če pravočasno ugotovimo okvaro podatkov, lahko na "standby" bazi transakcije uveljavimo samo do trenutka pred okvaro in s tem hitro zagotovimo prave podatke.

RPO in RTO sta pri teh rešitvah odvisna od nastavitev podvojevanja informacij o transakcijah, kot je prikazano v tabeli 3.3.

način podvojevanja	***RPO***	***RTO***
sinhrono	0	sekunde/minute
asinhrono	sekunde	sekunde/minute
pošiljanje log datotek	minute	minute/ure
zrcaljenje na nižjih nivojih	odvisno od tehnike	minute/ure

Tabela 3.3 Prikaz RTO in RPO v odvisnosti od načina prenosa informacij o transakcijah

Vsaka izvorna baza lahko informacije v transakcijah pošilja več "standby" bazam. To nam omogoča, da eno bazo držimo sinhronizirano z izvorno bazo, na drugi pa transakcije uveljavljamo z zamikom.

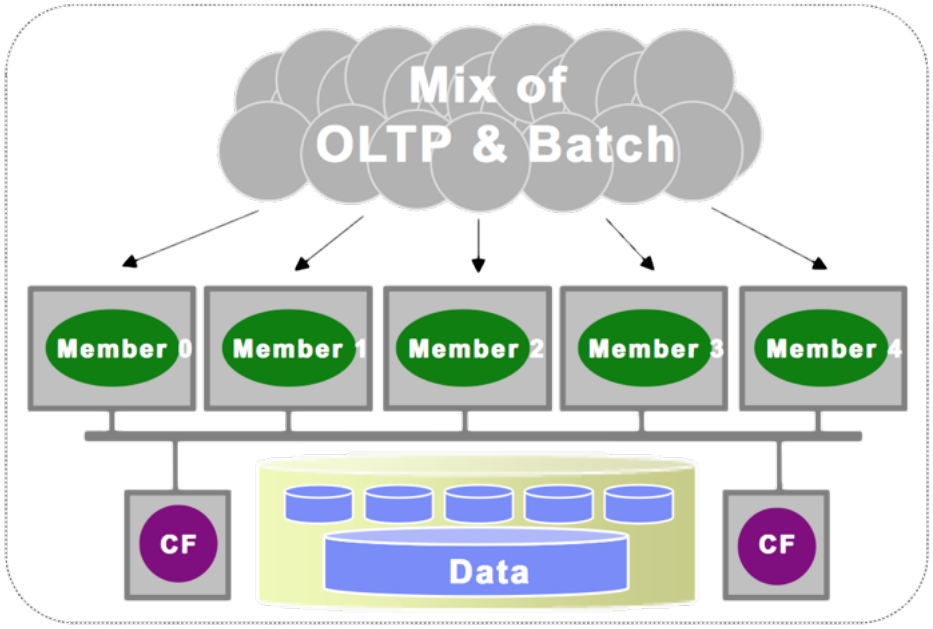
Postopek za prehod na "standby" bazo je lahko tudi avtomatiziran z uporabo skript ali namenskih aplikacij. Vsak prehod na drugo bazo za uporabnike pomeni prekinitev trenutne seje. Tovrstne težave se rešujejo z uporabo aktivno-aktivnih gruč, ki zagotavljajo hkratni dostop do istih podatkov na več strežnikih.

Aktivno-aktivne gruče

Aktivno-aktivna gruča je oznaka za gručo, v kateri več strežnikov istočasno ponuja iste podatke. Aktivno-aktivne gruče so primarno namenjene visoki razpoložljivosti na eni lokaciji. V primeru odpovedi enega strežnika so podatki razpoložljivi preko drugega in za uporabnika je prehod povsem transparenten[4]. Prav tako je razpoložljivost zagotovljena

[4]Oracle-ova rešitev zahteva uporabo prilagojenih aplikacij, IBM je to rešil na nivoju gruče.

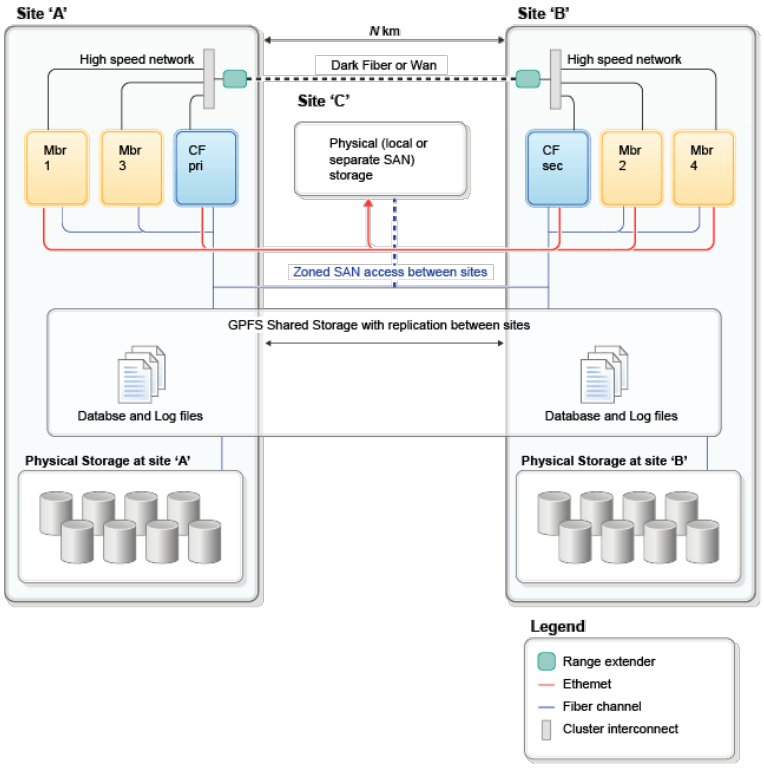
med nadgradnjami aplikacij, operacijskih sistemov ali strojne opreme. Edina neredundančna točka v teh rešitvah je diskovni sistem, ki pa jo znamo odpraviti. Strežniki za medsebojno komunikacijo uporabljajo nizkolatenčne povezave (slika 3.18).



Slika 3.18 Prikaz arhitekture pureScale [26].

V vsakem sistemu, kjer več uporabnikov hkrati bere in zapisuje podatke, je potrebno urediti veljavnost podatka. Klasični datotečni sistemi ne dovoljujejo, da bi imelo več strežnikov hkrati bralno-pisalni dostop, zato so za take rešitve neprimerni. Oracle uporablja rešitev za zapisovanje podatkov znotraj baze (ASM) [27], ostali proizvajalci pa uporabljajo datotečne sisteme, namenjene skupinski uporabi (GFS, GPFS, Sistina, Lustre, itd.). S tem zagotovimo, da je podatek na datotečnem sistemu vedno veljaven in konsistenten. Drugače je s podatkom v predpomnilnikih. Vsak strežnik uporablja svoj predpomnilnik za pohitritev dostopa do podatkov. Ker lahko različni odjemalci istočasno dostopajo do istega podatka na različnih strežnikih, je potrebno urediti tudi veljavnost podatka v predpomnilniku. DB2 za to uporablja t.i. "Cluster Facility" (CF), centralno entiteto, ki upravlja veljavnost strani v predpomnilnikih posameznih strežnikov. Vsa komunikacija gre preko CF, ki dovoljuje pisalni dostop posameznemu strežniku. Po zapisu podatka preko RDMA spremeni stran v predpomnilniku ostalih strežnikov v neveljavno. CF je podvojen zaradi zagotavljanja visoke razpoložljivosti [28]. Ostale podatkovne

baze za reševanje tega uporabljajo medstrežniško komunikacijo, ki poveča promet med strežniki in porabo procesorskih ciklov, ki se izkaže za pomanjkljivost pri večjih razdaljah med lokacijama.



Slika 3.19 Shema DB2 pureScale na dveh lokacijah [26].

Za zagotavljanje poslovanja v primeru izpada lokacije, lahko DB2 pureScale deluje tudi na večjih razdaljah (slika 3.19), vendar je zaradi Infiniband povezav med strežniki omejen na doseg Infinibanda (80 km[5]).

Pri tovrstnih rešitvah v primeru izpada strežnika, mrežne povezave ali diskovnega sistema nimamo izgube podatkov, poslovanje pa tudi ni prekinjeno.

3.3 Možnosti za znižanje stroškov hranjenja podatkov

V organizacijah, kjer so podatki aktualni kratek čas ali pa hranijo veliko podobnih kopij dokumentov, se je vredno vprašati, kako bi hranjenje takih podatkov finančno in performančno optimizirali. Določeni formati dokumentov so že kompresirani (predvsem mul-

[5]http://www.mellanox.com/related-docs/prod_long_haul_systems/MetroX_Brochure.pdf

timedijski), ostale pa je smiselno z uporabo različnih brezizgubnih tehnik kompresirati. Veliko podatkov se ponavlja, npr. imena, priimki, kraji. Količino zasedenega prostora takih podatkov lahko učinkovito zmanjšamo z uporabo deduplikacije. Do določenih podatkov dostopamo pogosteje, do nekaterih zahtevamo hiter dostop. Glede na vstopne parametre lahko podatke hranimo na različnih medijih in s tem pocenimo hrambo.

3.3.1 Optimizacija porabe prostora

Hranjenje podatkov je ena najdražjih storitev sodobnega informacijskega sistema. Cena na shranjeni podatek sicer pada, a je podatkov čedalje več, poleg tega pa poslovni procesi zahtevajo hitrejše dostope in večjo razpoložljivost. Shranjeni podatek zahteva še nekaj varnostnih kopij in mariskdaj dodatne hitro dostopne kopije na rezervnih lokacijah, vse to pa pomeni tudi večjo porabo energije za delovanje in hlajenje. Z uporabo kompresijskih tehnik lahko učinkovito zmanjšamo potrebo po prostoru in s tem neposredno vplivamo na TCO.

Deduplikacija in kompresija na datotečnih sistemih

Obstaja veliko datotečnih sistemov, ki omogočajo kompresijo podatkov. Nekateri omogočajo samo bralni dostop in za naš primer niso relevantni, zato se bomo osredotočili na datotečne sisteme, ki so primerni za uporabo pri fiktivnem naročniku.

Windows strežniki uporabljajo datotečni sistem NTFS, ki omogoča kompresijo posamezne datoteke s hitrim LZ77 algoritmom. Kompresija je transparentna do aplikacij. Vsaka aktivnost na kompresiranih datotekah gre preko knjižnic in potrebuje procesorski čas. Kompresija datotek, večjih od 30 GB je lahko neuspešna [29].

Operacijski sistem AIX uporablja datotečni sistem JFS ali JFS2. JFS omogoča kompresijo, vendar se ga zaradi omejitev skorajda ne uporablja več. Novejši JFS2 kompresije ne omogoča.

Linux omogoča uporabo različnih datotečnih sistemov. Transparentno kompresijo omogočata Btrfs in ZFS. Btrfs omogoča uporabo LZO (hiter, optimiziran za dekompresijo) ali Zlib (učinkovitejša kompresija). Kompresira na nivoju datoteke ali datotečnega sistema. Kompresira bloke velike 4kB, a ker vsak dostop do diska pomeni prenos 128kB podatkov, se vsakič dekompresira najmanj 128kB. Podpira tudi naknadno deduplikacijo na nivoju datotek [30]. ZFS za kompresijo lahko uporablja algoritme LZ4, LZJB, GZIP ali ZLE. Kompresijo se vklopi z ukazom na nivoju direktorija in vsi novo nastali doku-

menti se pred zapisom kompresirajo, obstoječi dokumenti pa ostanejo nedotaknjeni. ZFS podpira deduplikacjo na nivoju bloka. Za vsak blok zahteva 320 bajtov v tabeli vzorcev, ki se jo naslavlja ob vsakem dostopu do datotečnega sistema, zato je zaradi hitrosti dostopa praktično, da se nahaja v glavnem pomnilniku. ZFS v glavnem pomnilniku poleg tabele vzorcev hrani tudi t.i. Adaptive Replacement Cache (ARC) - rezerviran del glavnega pomnilnika, ki ga upravlja ZFS. Tabela vzorcev lahko zaseda največ 25% ARC, kar pomeni da rabimo štirikrat več RAM-a, kot je predvidoma velika tabela vzorcev. Na strežnikih, kjer hitrost dotopa ni ključnega pomena (npr. strežniki za varnostno kopiranje), se tabela vzorcev lahko prestavi tudi na L2ARC, drugonivojski ARC. Tega namesto na klasične diske postavimo na SSD ali Flash diske in tako ublažimo upad zmogljivosti [31]. Oba datotečna sistema podpirata tudi tehnologijo "copy-on-write".

Vklop kompresije lahko pomeni hitrejši dostop do podatkov, saj se v isti količini bralnih ali pisalnih zahtevkov prenese več podatkov. Prednost kompresije na datotečnih sistemih je v prilagodljovsti. Različne tipe datotek lahko kompresiramo z različnimi algoritmi, odvisno od zahtev lastnika. Vsaka kompresija ali deduplikacija podatkov pomeni, da je v primeru okvare datotečnega sistema potrebno več časa za okrevanje. Ta čas je potrebno upoštevati pri RTO.

Deduplikacija in kompresija v podatkovnih bazah

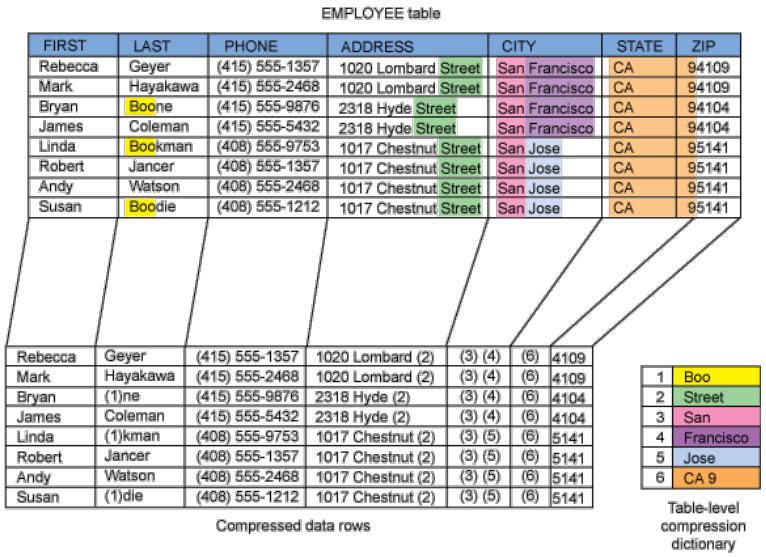
Podatkovne baze so model shranjevanja informacij, kjer se strukturirani podatki največkrat ponavljajo, zato uvedba metod za zmanjševanje količine zapisanih podatkov zelo učinkovito opravi svoje delo. Različne baze uporabljajo različne metode in algoritme za kompresijo, odvisno od mehanizma za zapisovanje podatkov. Baze kot so MongoDB, MariaDB, MySQL in Postgres, uporabljajo za zapisovanje namenski del programske kode, imenovan "storage engine", ki lahko podatke pred zapisom kompresira z uporabo različnih standardnih kompresijskih algoritmov. Ta koda je neizogiben člen med pomnilnikom in diskom in vsako branje in pisanje zahteva procesorski čas.

Oracle in DB2 uporabljata drugačen pristop in podatke zapisujeta direktno v tabele. Preden se podatki zapišejo, se lahko obdelajo z algoritmi, ki temeljijo na slovarjih (angl. *dictionary-based algorithms*). Algoritem deluje tako, da poišče ponavljajoče se vzorce in jih nadomesti s krajšim zapisom. Ker fiktivni naročnik uporablja DB2, se bomo v nadaljevanju osredotočili na to bazo. DB2 uporablja dve metodi stiskanja podatkov:

- Row compression in

- Adaptive row compression.

Row compression je mehanizem, ki deluje na nivoju tabele. Vsaka tabela v bazi ima lahko en slovar, imenovan slovar na nivoju tabele (angl. *table-level dictionary*), ki vsebuje kratke numerične ključe za ponavljajoče se vzorce. Primer delovanja prikazuje slika 3.20. Slovar je statičen, zato z dodajanjem podatkov v tabelo kompresijsko razmerje pada. Za osvežitev je potrebna reorganizacija tabele. Slovar tudi pri velikih tabelah redko preseže velikost 100kB[6]. Uporaba tega algoritma lahko celo pohitri delovanje baze, ker gre v eno stran več vrstic in je zato potrebnih manj vhodno-izhodnih operacij za branje podatkov.

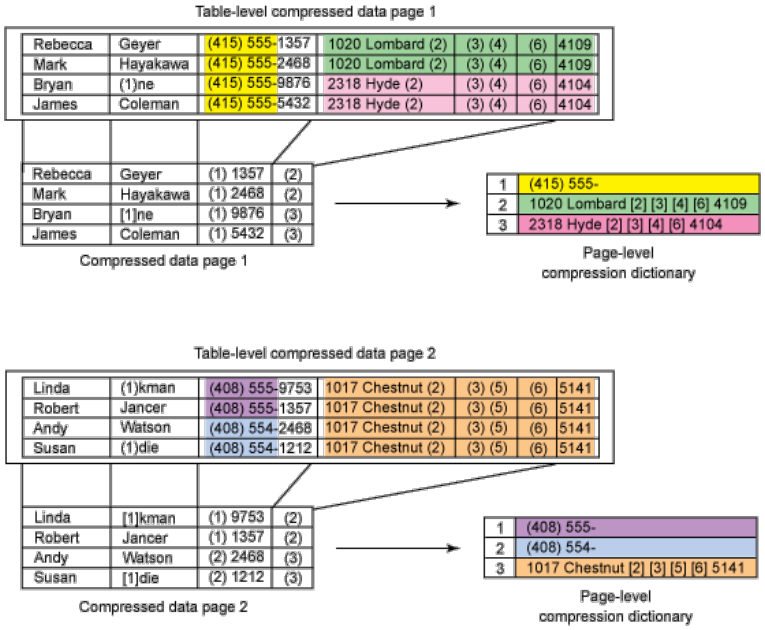


Slika 3.20 Prikaz delovanja Row compression [32].

Naprednejši mehanizem je Adaptive row compression. Izkorišča slovar na nivoju tabele in dodatno išče ponavljajoče se vzorce na nivoju strani (3.21). Vsaka stran dobi svoj slovar na nivoju strani (angl. *page-level dictionary*), ki ga proces "DB2 database manager" ustvari, ko se stran zapolni, ob večjih spremembah pa ga tudi posodobi. S tem odpade skrb za osveževanje vzorcev in kompresijsko razmerje s časom ostaja blizu optimalnega.

Tudi tovrstna kompresija zahteva procesorski čas, le da je ta precej manjši od klasičnih algoritmov, zato pa dodatno zaseda pomnilniški prostor, saj se z vsako odprto tabelo v

[6] http://www.ibm.com/support/knowledgecenter/api/content/nl/en-us/SSEPGG_10.5.0/com.ibm.db2.luw.admin.dbobj.doc/doc/c0056707.html

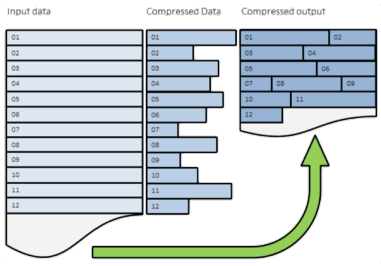


Slika 3.21 Prikaz delovanja Adaptive row compression [32].

pomnilnik prenese tudi slovar.

Deduplikacija in kompresija na nivoju diskovnih sistemov

Kompresijo podatkov ponujajo tudi diskovni sistemi. Najmanjša enota za kompresijo je običajno LUN. Uporablja se ena od izvedenk algoritma LZ, ker ob zadovoljivi kompresiji ponuja zelo hiter dostop [33]. S kompresijo se ukvarjajo procesorji na diskovnem sistemu, zato je potrebno uporabiti namenske procesorske kartice za kompresijo podatkov, če želimo obdržati odzivne čase in prepustnost. Pri tem je pomembna tudi verzija nameščene programske opreme[7].

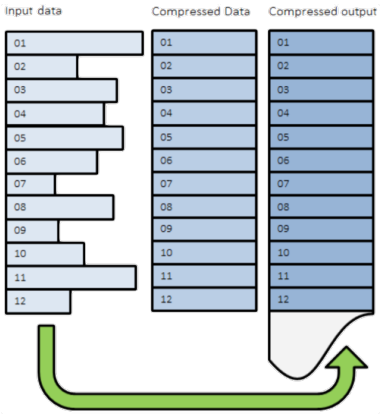


Slika 3.22 Tradicionalni način kompresije [11].

[7] http://www.bityard.org/blog/2014/11/23/ibm_svc_experiences_with_rtc

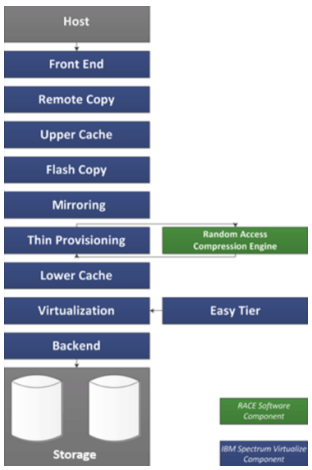
Algoritmi za kompresijo delujejo na principu iskanja ponavljajočih se vzorcev v določenem bloku. Podatek se razbije na koščke (angl. *chunk*) določene velikosti in znotraj njih se išče ponovitve. Kompresirani bloki so različno veliki in sekvenčno zapisani na disk (slika 3.22). Tak način ima eno pomanjkljivost: ob spremembi enega koščka je le-tega potebno prebrati, odkompresirati, spremeniti vrednost, nazaj skompresirati in zapisati. Kompresiran blok po spremembi ni nujno enako velik kot pred njo in s tem se povečuje razdrobljenost podatkov. Če je blok velik, to pomeni veliko vhodno-izhodnih operacij na disku, če je majhen, pa je kompresija neučinkovita.

IBM uporablja v diskovnih poljih drugačen pristop. Random Access Compression Engine (RACE) je patentirana implementacija LZ algoritma, ki deluje v realnem času in hkrati dovoljuje naključni dostop. Podatke razbije na koščke, ki odgovarjajo fiksni dolžini kompresiranega bloka, kot je vidno na sliki 3.23, kar omogoča učinkovito indeksiranje. Druga izboljšava je uporaba časovne kompresije (angl. *temporal compression*)



Slika 3.23 RACE kompresija [11].

namesto prostorske (angl. *location-based compression*). Časovna kompresija predvideva, da podatke, ki pridejo v vrsti, pošlje ista aplikacija. Ti podatki so istega tipa, s tem pa je večja verjetnost ponavljajočih se vzorcev in posledično je kompresija učinkovitejša [33]. Dodatno povečanje učinkovitosti omogoča še mehanizem za vnaprejšnje odločanje o kompresiji (angl. *predecided mechanism*), ki vzorec koščka analizira in se odloči, ali bo sploh kompresiran in s katerim algoritmom. RACE je možno tudi paralelizirati. Implementiran je na nivoju "Thin provisioning" (slika 3.24), zato je potrebno stalno spremljanje zasedenosti in ustrezno reagiranje, preden zmanjka prostora. Strežnik dobi potrditev o zapisu takoj, ko diskovni sistem sprejme podatek v zgornji predpomnilnik. S tem



Slika 3.24 Spectrum Virtualize sklad [11].

zmanjšamo latenco, arhitektura diskovnega sistema pa mora omogočiti, da se podatki v predpomnilniku vedno tudi zapišejo na diske (baterije, NVRAM).

Druga tehnika za zmanjševanje količine zapisanih podatkov je deduplikacija. Lahko se izvaja med zapisovanjem ali pa naknadno na že zapisanih podatkih v časovnih intervalih. IBM uporablja deduplikacijo v programskih in strojnih rešitvah za varnostno kopiranje podatkov, konkurenčna podjetja pa jo uporabljajo tudi pri ostalih diskovnih sistemih. Deduplikacija prinaša prednost pri količini zapisanih podatkov, potrebno pa se je zavedati, da se število dostopov do diska precej poveča - odvisno od velikosti bloka za deduplikacijo.

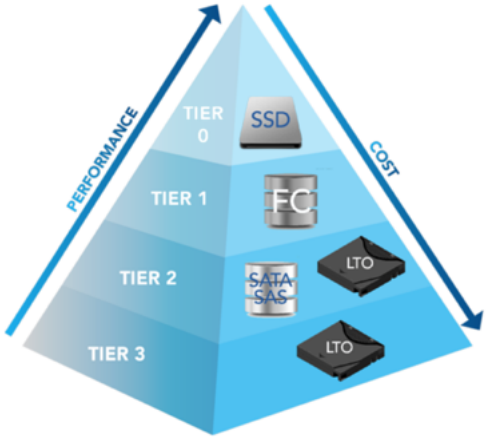
Kompresija in deduplikacija na diskovnih sistemih sta povsem transparentni do aplikacij. Prednost je v upravljanju iz enega mesta in obdelava brez obremenitve glavnega procesorja.

3.3.2 Hierarhični model shranjevanja

V vsakem prostoru za shranjevanje podatkov hote ali nehote ustvarimo nekakšno hierarhijo. Večkrat uporabljeni podatki so lažje dostopni kot arhivi, ki so običajno nekje zadaj, zloženi na težje dostopen prostor ali v obliki, ki zahteva dodatno obdelavo pred uporabo. Enak princip lahko uvedemo tudi na elektronsko področje. Ideja je, da pogosto uporabljane podatke, imenovane tudi aktiva (angl. *online storage*), hranimo na medijih z naključnim ali direktnim dostopom in so uporabnikom dostopni takoj. Podatke, ki jih uporabljamo redkeje, imenujemo arhiva (angl. *nearline storage*). Dostopni so lahko z

zamikom, ker jih lahko hranimo tudi na medijih s sekvenčnim dostopom. Dostop je za uporabnika transparenten. Na hitre medije se zapišejo kazalci in v primeru dostopanja do podatka na arhivi se le-ta prenese v aktivo. Podatki se iz hitrejših medijev avtomatsko prenašajo na počasnejše, ko zapolnimo določen prostor. Običajno se podatki za selitev na arhivo izberejo z LRU algoritmom, možno pa je izbrati tudi drugačne pogoje.

Prihranki pri takem načinu shranjevanja podatkov so občutni, saj je razlika v ceni na količino lahko večstokratna.



Slika 3.25 Prikaz večnivojskega HSM [34].

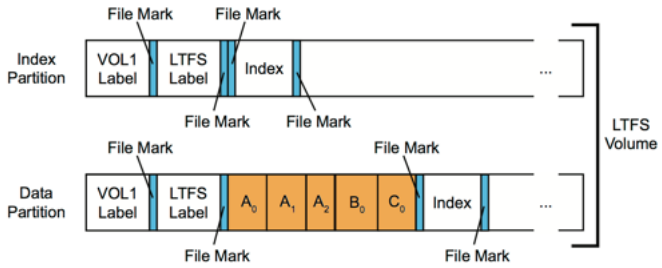
V času osnovanja HSM (IBM 3850, 1974[8]) so bile razmere precej drugačne od današnjih. Diski so bili relativno majhni in zelo dragi, magnetni trakovi pa so ponujali veliko prostora za sprejemljivo ceno. Z razvojem tehnologij in nižanjem cene se je HSM počasi prelevil v današnje stanje, ko je HSM večnivojski, uporablja pa lahko tako Flash kot trde diske in magnetne trakove (prikazano na sliki 3.25). HSM je možno implementirati na strojnih in aplikativnih nivojih.

HSM za datotečne sisteme

Na nivoju datotečnih sistemov lahko uporabljamo hierarhični model shranjevanja z uporabo namenske opreme ali pa datotečnih sistemov, ki omogočajo razširitev naslovnega prostora na LTFS in s tem uporabo poceni 3592 ali LTO trakov za hrambo velikih količin podatkov.

[8] https://www-03.ibm.com/ibm/history/exhibits/storage/storage_3850.html

LTFS je aplikacija protokola, ki vsebino magnetnega traku prikaže kot da bi bil disk. Deluje na trakovih, ki podpirajo particioniranje. Ena particija se uporablja za kazalo ("index partition") in vsebuje opisne podatke datotečnega sistema (angl. *metadata*), na drugi particiji pa je zapisana vsebina datotek (slika 3.26). Magnetni trak je še vedno sekvenčni medij: vsebina se vedno zapisuje na konec in brisanje pomeni samo umik kazalca. Z LTFS je poenostavljeno upravljanje z vsebino na traku, poleg tega pa omogoča branje traku na vsakem sistemu, ne glede na to, kje je bil zapisan. Celotna specifikacija protokola je na razpolago na spletnih straneh SNIA [35].



Slika 3.26 Prikaz LTFS formata [36].

HSM skladno s politiko umika datoteke iz diskov na trakove. Na diskih pusti delček datoteke (angl. *stub file*), ki vsebuje vse opisne podatke. Uporabniki datoteko še vedno vidijo kot lokalno, le čas dostopa do te datoteke se podaljša. Zaradi sekvenčnega medija se dostopni čas spreminja odvisno od lokacije zapisa. Z uporabo LTO-6 trakov je običajno pod 60 sekund [36]. Politike za umik datotek se lahko prilagodijo uporabniku, lahko so enostavne (LRU, starost dokumenta) ali pa zelo kompleksne (tip ali velikost dokumenta, vezano na starost in uporabnika, itd). IBM z rešitvijo SONAS omogoča opisovanje politik z jezikom, podobnim SQL. Ob periodičnih pregledih datotečnih sistemov se datoteke razvrsti po pravilu prvega ujemanja. Vsaka datoteka je lahko samo v eni politiki, zato je pomemben vrstni red politik.

Čeprav lahko s HSM datoteke umikamo na drugo lokacijo, take rešitve ne moremo uporabiti za varnostno kopijo. S pomočjo kazalcev nazaj sicer lahko dostopamo do prejšnje verzije dokumenta, ne moremo pa dostopati do pobrisanih datotek, ker HSM datoteke premika in ne podvaja.

Na voljo imamo več načinov implementacije HSM:

- aplikacija na datotečnem strežniku,

- diskovni sistem, ki ponuja datotečni (angl. *file-level*) dostop preko omrežnih protokolov,
- naprava, ki omogoča umikanje datotek na obstoječih diskovnih strežnikih.

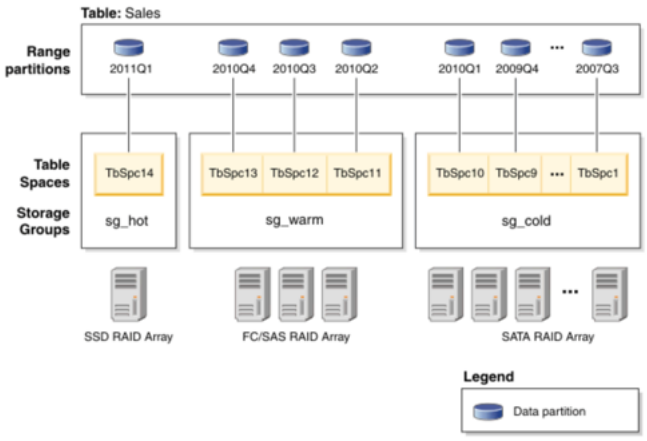
Osnovni princip je povsod enak, razlika je le na nivoju, ki opravlja HSM.

Hierarhija v podatkovnih bazah

Particioniranje podatkovne baze omogoča, da podatke v tabelah ločimo po izbranem ključu in jih shranimo na različnih lokacijah na diskih. Že tukaj lahko uporabimo različno zmogljive oz. drage diske, v kombinaciji s HSM na nivoju datotečnih sistemov pa lahko posamezne particije preselimo tudi na trak. Integracija podatkovne baze s programsko opremo za HSM uporabnikom ponuja transparentnost. Posamezne particije lahko aktiviramo po potrebi, kar prinaša še dodatne ugodnosti:

- zmanjšuje potrebo po pomnilniku,
- omogoča kratek RTO za kritične podatke,
- pohitri izvajanje obdelav,
- poenostavi ILM, če obstaja zakonska zahteva po uničenju podatkov.

Pomanjkljivost tega načina je, da so podatki statično razdeljeni. Med prestavljanjem podatkov iz ene particije v drugo je del podatkov nedostopen.



Slika 3.27 Primer razdelitve podatkov glede na temperaturo [10].

"Multi-temperature data management" je izboljšava prej opisanega načina za razdelitev podatkov. Omogoča razvrščanje podatkov glede na njihovo temperaturo medtem

ko so podatki normalno dostopni. Prostor za podatke razdelimo v podatkovne skupine in znotraj teh definiramo "tablespace". Podatkovne skupine naredimo na ustreznih diskih skladno s temperaturo podatkov - vroči podatki so na hitrih diskih, mrzli na počasnih (slika 3.27). Selitev podatkov med podatkovnimi skupinami je transparentna do uporabnikov. Izvaja se na zahtevo upravljalca, možno je nastaviti prioriteto in omejiti vpliv na odzivnost baze. Med izvajanjem zahtevnih obdelav jo je možno tudi začasno ustaviti.

"Tiering na nivoju diskovnih sistemov"

Hierarhično zasnovo shranjevanja podatkov na diskovnih poljih imenujemo "tiering". Z enim krmilnikom krmilimo različno hitre medije z naključnim in direktnim dostopom in podatke selimo na osnovi LRU algoritma. Enake diske združimo v RAID polja, nato pa RAID polja različnih diskov združimo v eno entiteto, imenovano "pool", na kateri kreiramo LUN-e. LUN-i so razdeljeni na bloke velikosti 64kB (angl. *extent*). V "pool-u" so lahko eden, dva ali trije razredi RAID polj. IBM jih imenuje flash, enterprise in nearline. Diski so v razrede uvrščeni avtomatsko glede na tip diska. Možno je tudi ročno uvrščanje v primerih, ko je en razred na zunanjem diskovnem sistemu ali pa imamo več diskov iz istega razreda. Razrede diskov prikazuje tabela 3.4.

vrsta diskov	***razred***
SSD	flash
SAS HDD 15k	enterprise
SAS HDD 10k	enterprise
SATA 7k2	nearline
Zunanja diskovna polja	enterprise

Tabela 3.4 Razvrstitev diskov v razrede.

Programska koda, imenovana Easy Tier, je del virtualizacijskega nivoja sklada Spectrum Virtualize (slika 3.24). Easy Tier ves čas spremlja pogostost dostopa do blokov in latenco in na osnovi teh podatkov pripravi plan migracije blokov. Vroče bloke transparentno preseli na višji, mrzle pa na nižji razred. Če "pool" vsebuje samo en razred diskov na več RAID poljih, Easy Tier na podoben način izenačuje obremenjenost posameznih RAID polj.

Poleg migracije blokov med razredi Easy Tier tudi umika sekvenčno dostopane bloke na razred nearline. S tem obremeni tudi SATA diske z njim primerno obremenitvijo [11].

Netapp uporablja bloke velikosti 4kB, kar pomeni, da enako učinkovitost dosežemo že z manjšo količino Flash diskov. Uporablja dva razreda, enega za kapaciteto in enega za zmogljivost. Poleg omenjene uporabe omogoča še uporabo Flash diskov kot predpomnilnik za podatke na diskih z uporabo klasičnih algoritmov. S tem, ko se vsak dostopani blok prepiše v Flash, lahko pospešimo tudi kratke obremenitve. Sekvenčna branja so izvzeta iz tega postopka in se vedno izvajajo direktno iz diskov. Izognemo se tudi prepisovanju manj uporabljanih podatkov nazaj na diske in tako zmanjšamo količino internega prometa [37].

EMC ponuja enaki možnosti kot NetApp. Razlika je v tem da je najmanjša enota za migriranje na Flash diske 1GB, če Flash uporablja kot predpomnilnik, pa 4kB [38].

Slabost uporabe Flash diskov kot predpomilnika je v življenjski dobi oz. številu prepisov pomnilniških celic.

4 Rešitev

V pričujočem poglavju bomo povzeli prednosti in slabosti posameznih tehnoloških možnosti in jih umestili v zahteve fiktivnega naročnika. Tehnične službe naročnikov imajo večkrat težave z upravičevanjem določenih rešitev, ker podjetje nima izdelane strategije in določenih postopkov za ravnanje v primeru dolgo trajajočega reševanja na informacijski infrastrukturi. Reševanje situacij poteka ad-hoc, veliko časa pa se porabi za čakanje na odziv podpornih služb. Za upravičevanje določenih rešitev za doseganje visoke razpoložljivosti je potrebno rešitev pripraviti v skladu z načrtom neprekinjenega poslovanja. Le-tega mora potrditi vodstveni kader, ki pa se velikokrat ne zaveda, kaj za delovni proces pomeni ena ura izpada.

4.1 Osnutek načrta neprekinjenega poslovanja

Naročnik želi z rešitvijo zagotoviti razpoložljivost podatkov skladno s tabelo 2.1. Zaradi naročnikove izpostavljenosti lahko take zahteve izpolnimo le, če vzpostavimo drugo lokacijo za ključne elemente informacijskega sistema.

Za informacijski sistem je potrebno pripraviti ustrezen načrt neprekinjenega poslova-

nja. V distribuiranem sistemu, kakršnega ima naročnik, je potrebno v varovanje podatkov in zagotavljanje klimatskih pogojev vlagati precej truda, ima pa ta sistem tudi prednost, da v primeru izpada enega dela sistema ostali deli delujejo nemoteno. Kljub temu so koristi centraliziranega sistema večje od možnih težav, sploh če poskrbimo za ustrezno razpoložljivost v primeru okvar. V primerjavi z distribuiranim sistemom s centralizacijo in uporabo virtualizacije povečamo izkoristke in na ta način močno zmanjšamo TCO:

- potrebujemo manj strojne opreme in opreme za klimatizacijo,
- poceni in poenostavi se vzdrževanje,
- zmanjša se potreba po prostoru,
- zmanjša se poraba energije,
- znižajo se stroški za varovanje podatkov,
- znižajo se stroški licenc,
- delovni pogoji za upravitelje so ugodnejši.

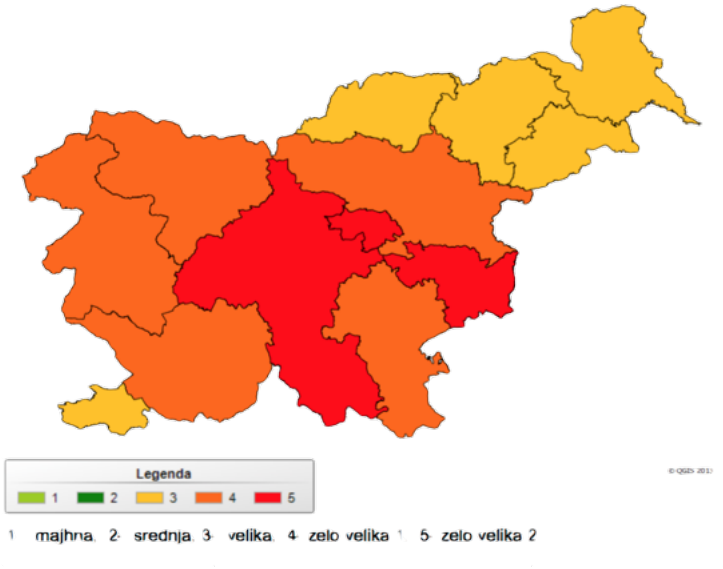
Posledično se zmanjša verjetnost okvar in tudi za nepridiprave je manj vstopnih točk v sistem. V tabeli 4.1 so prikazane verjetnosti pozameznih groženj in potrebne rešitve za zagotavljanje razpoložljivosti informacijskih storitev.

grožnja	***verjetnost***	***stopnja ranljivosti***	***razlog za stopnjo ranljivosti***
okvara podatkov	nizka	nizka	varnostna kopija
odpoved diska	visoka	nizka	RAID polja
odpoved vhodno/izhodne naprave	nizka	srednja	podvojeni adapterji, malo adapterjev zaradi virtualizacije
odpoved strežnika	nizka	nizka	visoko razpoložljive gruče, selitev med delovanjem
odpoved diskovnega sistema	nizka	nizka	replikacija podatkov
odpoved na komunikacijski poti	srednja	nizka	podvojene poti, veliko opreme
izpad lokacije	nizka	nizka	visoko razpoložljive gruče, replikacija podatkov
bolezen/poškodba skrbnika	visoka	srednja	razmeroma majhna skrbniška ekipa
vdor v omrežje	nizka	srednja	podatki so dostopni veliko ljudem, omejene pravice zapisovanja

Tabela 4.1 Analiza in riziko groženj.

Ker je v Sloveniji zdravstvenih ustanov veliko, verjetnost za uničenje posamezne ustanove pa nizka, nima smisla podvajati drage opreme za zagotavljanje delovnega procesa, saj lahko paciente preusmerimo drugam. Rezervno lokacijo torej iščemo samo za zagotavljanje informacijske podpore in dostopnost podatkov drugim zdravstvenim ustanovam, vključenim v sistem zNET. Za varovanje pred poplavo ali požarom je dovolj izbrati drugo lokacijo znotraj kampusa. Rezervno lokacijo je torej potrebno umestiti glede na potresno ogroženost Slovenije, ki jo prikazuje slika 4.1. Slovenija je potresno ogrožena država, saj

večina države spada v območje VI. - VIII. stopnje EMS, obenem pa jo seka veliko tektonskih prelomnic, zato je verjetnost potresa, ki bi prizadel veliko površino relativno majhna. Rezervno lokacijo lahko torej iščemo relativno blizu, kar pozitivno vpliva na latenco in ceno povezave. Strošek rezervne lokacije je možno še dodatno znižati z medsebojnim nudenjem prostora. Ob izpadu lokacije je pomembno, da se zagotovi čimbolj enostaven



Slika 4.1 Karta potresne ogroženosti regij v Sloveniji [39].

prenos storitev na rezervno lokacijo. Poleg tega ima naročnik tudi zelo omejeno podporno službo za informatiko zato je priporočljivo, da so postopki za prenos različnih storitev čimbolj poenoteni in stestirani. Poskrbimo za avtomatsko sinhronizacijo podatkov in delovanje visoko razpoložljivih gruč iz primarne lokacije razširimo na rezervno lokacijo. S tem odpade potreba po ročnem delu v primeru napake.

Rešitev mora zaradi zahtev za RTO ponujati možnost prenosa podatkov med različnimi diskovnimi sistemi med delovanjem, saj si ne morejo privoščiti dovolj dolge prekinitve za kopiranje podatkov.

4.2 Povečanje razpoložljivosti podatkov

Glede na načrt neprekinjenega poslovanja predlagamo uvedbo virtualizacijskega nivoja na diskovnih sistemih. Naročnik ima veliko različnih diskovnih sistemov, ki jih bo na ta način lahko izkoristil in še naprej uporabljal, poleg tega pa virtualizacija na diskovnih sistemih

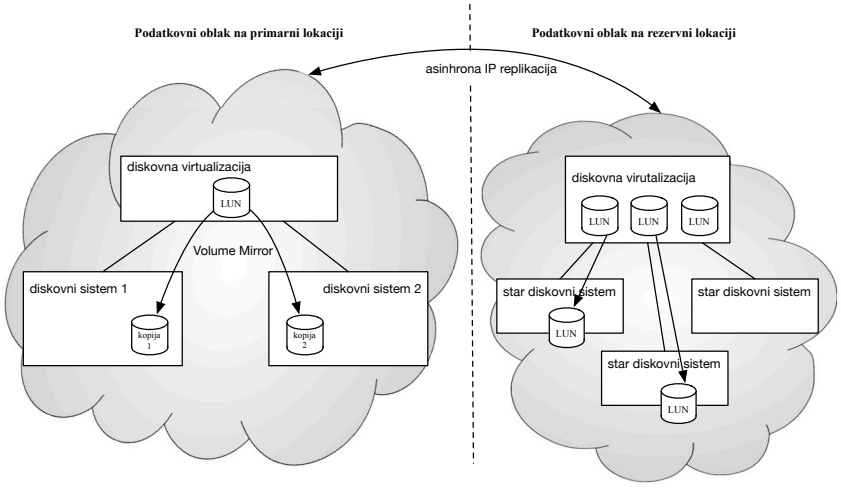
omogoča prenašanje podatkov med različnimi diskovnimi sistemi medtem ko so podatki na voljo uporabnikom. To nam obenem pušča svobodo pri odločitvah v prihodnosti, ker nas ne veže na določenega proizvajalca. Obstoječe diskovne kapacitete, ki trenutno prostorsko zadovoljujejo potrebam, bi uporabili na rezervni lokaciji, na primarni lokaciji pa bi postavili novo diskovje.

Rezervno lokacijo bi glede na sliko 4.1 iskali v sosednji regiji, kar pomeni oddaljenost vsaj 70 km. To pomeni, da bo latenca do oddaljene lokacije vsaj $350\mu s$ v eno stran.

Iz podanega letnega prirastka podatkov lahko izračunamo potrebno pasovno širino linije med obema lokacijama. Večina od 15 TB podatkov se generira med ponedeljkom in soboto med 7. in 19. uro. V 313 dneh se tako povprečno zgenerira 4 GB podatkov na uro, ki jih je potrebno prenesti na rezervno lokacijo. Za tako količino podatkov potrebujemo stalno pasovno širino 9,1 Mbps. Taka linija potrebuje na 70 km za sinhronizacijo enega bloka podatkov, velikega 4kB,

$$700\mu s + \frac{4 \cdot 8 \cdot 2^{10}}{9,1 \cdot 2^{20}} \simeq 4,13ms. \tag{4.1}$$

Vzemimo kot primer sliko, veliko 20 MB. Ta bi se po tej liniji prenašala 17,6s. Diskovni



Slika 4.2 Arhitektura rešitve za povečanje razpoložljivosti podatkov.

sistemi z veliko diski so sposobni brez težav dosegati hitrosti nekaj 100 MB/s. To pomeni, da prej kot v pol minute zapiše tako količino podatkov, da presega dovoljeni RPO. Poleg novih podatkov je potrebno upoštevati še spremembe na starih podatkih, zato je potrebno

zmogljivost linije ocenjevati drugače. Slika bi se preko idealne 100 Mbps linije prenašala 1,6007s, preko 1Gbps pa 0,1607s. Količine sprememb v enem dnevu ne poznamo, zato bo potrebno zahtevano pasovno širino linije oceniti. Tudi če je sprememb toliko kot novo ustvarjenih dokumentov, smo jih sposobni pospraviti na rezervno lokacijo znotraj predvidenega RPO po liniji z zmogljivostjo 200-300 Mbps. To pomeni, da med lokacijama ne potrebujemo drage FC povezave (ki se meri v večkratnikih Gbps), ampak bo dovolj IP povezava.

Na primarni lokaciji naročnik uporablja visoko razpoložljive gruče za zaščito pred napakami na strežnikih, želi pa tudi zaščito podatkov pred napakami na diskovnih sistemih. Zato za primarno stran ponudimo rešitev z dodatnim nivojem diskovne virtualizacije, ki z redundančno zasnovo omogoča virtualizacijo več zunanjih diskovnih sistemov in hkrati upravljanje iz enega mesta. Z uporabo Volume Mirror zagotovimo zaščito pred izpadom diskovnega sistema na primarni lokaciji, kot je prikazano na sliki 4.2.

Kljub relativno kratki razdalji je latenca že močno opazna, še posebej pri zahtevah po hitrih transakcijah. Zaradi omogočanja hitrega odziva aplikacij bi uporabili asinhrono replikacijo podatkov, ker ob dovolj zmogljivi liniji tudi asinhrona replikacija zagotavlja, da zahtevan RPO ne bo presežen. Tabela 4.2 prikazuje povzetek možnih rešitev za replikacijo podatkov z ocenjenimi razredi RTO in RPO glede na do sedaj sprejete odločitve:

- rezervna lokacija,
- asinhrona replikacija,
- IP povezava med lokacijama,
- podvojeni podatki na primarni lokaciji,
- avtomatski prehod na rezervno lokacijo,
- čimbolj poenoteno upravljanje.

Replikacija na operacijskih sistemih je neprimerna zaradi razdrobljenega upravljanja, “split mirror” je primeren predvsem za občasno kopiranje podatkov, aktivno-aktivne gruče pa zahtevajo Infiniband povezave med lokacijama. Vsi ostali načini podvajanja podatkov, omenjeni v tretjem poglavju, zahtevajo FC povezave med lokacijama. Global Mirror ponuja krajši RPO kot GMCV. Naročnikovim zahtevam zadostujeta oba, zato za repliciranje podatkov na datotečnih sistemih predlagamo uporabo GMCV, ki ne potrebuje dodatne opreme za pošiljanje FC protokola preko IP opvezave. Zahtevam ustreza tudi replikacija na nivoju podatkovnih baz, ki pa zahteva dodatno udejstvovanje, a po-

področje	*način replikacije*	*RTO*	*RPO*	*skladnost z zahtevami*
operacijski sistem	zrcaljenje z LVM	sekunde/minute	sekunde	ne
	prestrezanje sprememb v datotekah	minute	sekunde/minute	ne
diskovni sistem	Global Mirror	sekunde	sekunde	da
	Global Mirror with Change Volumes	sekunde	2 x perioda izvajanja	da
podatkovne baze	split mirror	minute	perioda izvajanja	ne
	zrcaljenje	sekunde/minute	sekunde	da
	replikacija informacij o transakcijah	minute/ure	minute	da
	aktivno-atkivne gruče	0	0	ne

Tabela 4.2 Skladnost tehnoloških rešitev z zahtevami.

nuja več varnosti. Glede na naročnikov kader in pojavnost napak na podatkih je potrebno oceniti, ali se kljub temu splača podatkovne baze podvajati s pošiljanjem log datotek in uveljavljanjem informacij o transakcijah z zamikom.

4.3 Optimizacija shranjevanja podatkov

Kompresija podatkov na nivoju podatkovne baze prinese precejšnje učinke, saj se po izkušnjah potrebni prostor za shranjevanje zmanjaša tudi do 80%. Le malo nižji odstotek prinese tudi kompresija LUN-ov na diskovnih sistemih, medtem ko deduplikacija na diskovnih sistemih prinese okrog 50% prihranka na datotečnih sistemih in 10-20% na podatkovnih bazah. Prednosti in slabosti posameznih rešitev so naštete v tabeli 4.3.

področje	*rešitev*	*prednosti*	*slabosti*
datotečni sistem	kompresija	možnost kompresiranja posameznih datotek	razdrobljenost
	deduplikacija	učinkovita	samo na datotečnih sistemih za Linux
diskovni sistem	kompresija	ne obremenjuje procesorjev na strežnikih	replicirajo se nekompresirani podatki
	deduplikacija	upravljanje iz enega mesta	slabša učinkovitost od kompresije
podatkovne baze	kompresija	replicirajo se kompresirani podatki	razdrobljenost, draga licenca

Tabela 4.3 Primerjava načinov za zmanjšanje količine zapisanih podatkov.

Če so v igri velike razdalje, kjer je težava količina prenešenih podatkov, je kompresija na podatkovnih bazah smiselna rešitev, na kratkih razdaljah pa cena kompresije ne upraviči stroška za širšo pasovno širino linije. Tehnologija vnaprejšnjega odločanja omogoča časovno učinkovito kompresijo LUN-ov z datotečnimi sistemi kljub veliki količini že kompresiranih podatkov. Zaradi upravljanja iz enega mesta, nižje cene in enostavnosti

predlagamo uporabo kompresije na diskovnih sistemih.

Dodatna stroškovna optimizacija je možna z uporabo cenejših medijev za hrambo redkeje dostopanih podatkov. Naročnik ima vse podatke dostopne preko datotečnih sitemov, zato bi bila na prvi pogled ustrezna rešitev:

- HSM na datotečnih sistemih,
- "Multi-temperature data management" na podatkovnih bazah.

IBM Storwize V7000 Unified ponuja datotečni (za dostop do datotečnih sistemov) in blokovni (diski za podatkovne baze) dostop do podatkov v eni napravi. Omogoča diskovno virtualizacijo in replikacijo podatkov, z vgrajeno rešitvijo Active Cloud Engine® in zunanjim Spectrum Protect strežnikom tudi migracijo podatkov na magnetne trakove, zato bi ga lahko na obeh lokacijah uporabili kot nivo diskovne virtualizacije (primerjaj s sliko 4.2) [40]. Naročnik do sedaj na podatkovnih bazah ni uporabljal kompresije in particioniranja, zato so zadostovale cenejše Workgroup licence. Zaradi zahtev po razpoložljivosti podatkov bi morali uporabiti "Multi-temperature data management", ki pa je omogočen šele v skoraj trikrat dražji Advanced Workgroup licenci[1]. Ta licenca je zahtevana tudi za uporabo kompresije.

Naročnikova tračna knjižnica ni dovolj velika, da bi zmogla to dodatno obremenitev. Za shranjevanje podatkov, starejših od štirih let bodo v naslednjih petih letih potrebovali

$$kolicina\ podatkov + predvidena\ rast \cdot stevilo\ let = 240TB + 15TB \cdot 5 = 315TB \quad (4.2)$$

prostora. Z LTO-5 tehnologijo bi za to potrebovali

$$\frac{kolicina\ podatkov}{kapacitetatraku} = \frac{315TB}{1,5TB} = 196 \quad (4.3)$$

trakov. Za uporabo magnetnih trakov kot medija za shranjevanje mrzlih in mirujočih podatkov bi morali tračno knjižnico nadgraditi ali pa jo zamenjati.

Kljub posameznim ugodnostim te rešitve je kot celota ekonomsko bolj upravičeno izvajanje HSM znotraj diskovnega sistema. Potrebujemo sicer večjo količino diskov, vendar lahko za mrzle in mirujoče podatke uporabimo velike, a relativno počasne diske. Dostop do podatkov na teh diskih je hitrejši kot do podatkov na magnetnih trakovih. Tudi cenovno je pri naročnikovi količini podatkov rešitev z diski ugodnejša.

[1] http://www.ibm.com/support/knowledgecenter/?lang=en#!/SSEPGG_10.5.0/com.ibm.db2.luw.licensing.doc/doc/r0053238.html

vrsta podatkov	*izračunana predvidena količina*	*računska količina*	
		z rezervo	*s kompresijo*
vsi	$300TB + 5let \cdot 15TB/leto = 375TB$	400TB	340TB
kompresirani z zapisom	$375TB \cdot 60\% = 225TB$	240TB	240TB
vroči	$15TB/leto \cdot 6mesecev = 7,5TB$	8TB	7TB
topli	$15TB/leto \cdot 42mesecev = 52,5TB$	53TB	45TB
mrzli in mirujoči	$375TB - 7,5TB - 52,5TB = 315TB$	340TB	288TB

Tabela 4.4 Izračun predvidene količine podatkov.

4.4 Izračun in izbira ustrezne opreme

Za zadovoljitev naročnikovih potreb najprej izračunamo količino posameznih podatkov ob konzervativni predpostavki, da kompresija prinese 60 odstoten prihranek prostora. Razdelitev podatkov po količinah je predstavljena v tabeli 4.4.

Za hierarhijo shranjevanja predlagamo uporabo Easy Tier. S to rešitvijo odstranimo potrebo po starostni opredelitvi podatkov in uvedemo delitev podatkov glede na dostopanost. Osnova za izračun je še vedno časovna delitev iz tabele 2.1, upoštevane pa so prednosti hierarhične metode shranjevanja. Predlagamo uvedbo tristopenjskega HSM:

- za vroče podatke namenimo SSD diske,
- tople podatke hranimo na SAS diskih,
- vse ostalo hranimo na SATA diskih.

Easy Tier bo poskrbel, da bodo pogosto dostopani podatki na SSD diskih, ostali diski pa še vedno ponujajo hitrejše dostopne čase od zahtevanih. Zaradi količine podatkov in uporabe Easy Tier je smiselno, da uporabimo dokaj velike SAS in SATA diske, ker so na količino cenejši, potrebujemo manj prostora v omarah in hkrati zmanjšamo porabo energije. Za zagotavljanje kratkih okrevalnih časov v primeru odpovedi diskov predlagamo uporabo “Distributed RAID”. Na primarni lokaciji bodo podatki podvojeni na drug diskovni sistem, zato je smiselna uporaba RAID-5 polj. Iz tega stališča bi bilo na rezervni lokaciji primerno uporabiti RAID-6 polja, a je potrebno oceniti, ali so glede na pogostost uporabe rezervne lokacije sprejemljivi višji stroški. Za postavitev RAID-6 namreč potrebujemo 12% več diskov. Na SAS in SATA diskih predlagamo uporabo RAID-5 polj v organizaciji 7+1 zaradi ujemanja s performančo optimizacijo bralnega algoritma v IBM Spectrum Virtualize. Za zadostitev potrebam potrebujemo konfiguracijo, prikazano v

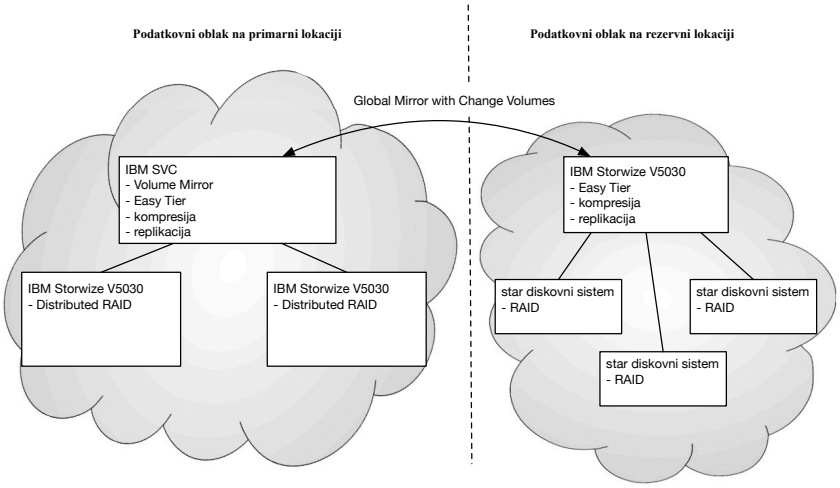
tabeli 4.5.

vrsta	*velikost diska*	*število diskov za*			*vsota*	*uporabna kapaciteta*
		podatke	*pariteto*	*rezervo*		
SSD	1,6TB/2,5”	4	1	1	6	6,4TB
SAS	1,2TB/2,5”	35	5	4	44	42TB
SATA	6TB	49	7	4	60	294TB

Tabela 4.5 Izračun potrebne količine diskov.

Konfiguracija diskov na rezervni lokaciji zaradi asinhrone replikacije nima vpliva na zmogljivosti na primarni lokaciji, prav tako Easy Tier ne vpliva na replikacijo, ker se le-ta izvaja na višjem nivoju (slika 3.24). Količina prostora na starih diskovnih poljih ne zadošča za nadaljnih 5 let. Tudi vsi stari diskovni sistemi niso primerni za uporabo na rezervni lokaciji, ker diskovna virtualizacija zahteva FC povezavo. Diske, potrebne za pokritje razlike, vgradimo v kontrolni sistem za diskovno virtualizacijo.

Za diskovno virtualizacijo na primarni strani predlagamo IBM SVC. Modularna sestava mu omogoča strojne in programske nadgradnje brez prekinitve delovanja ter za uporabnike transparentno migracijo podatkov med podrejenimi diskovnimi sistemi. IBM SVC ima kljub temu eno pomanjkljivost: programska oprema na nivoju diskovne virtualizacije ni podvojena in v primeru napake izgubimo dostop do vseh diskov. Stopnjo



Slika 4.3 Konfiguracija diskovnih sistemov rešitve za povečanje razpoložljivosti podatkov.

ranljivosti lahko zmanjšamo tako, da uporabimo enako ali podobno rešitev za diskovno virtualizacijo še na rezervni lokaciji in/ali na podrejenih diskovnih sistemih. S tem pridobimo možnost testiranja programske opreme in preverjanja postopkov nadgradnje. Strojna kompresija podatkov in predpomnilnik, vgrajen v SVC, omogočata uporabo manj zmogljivih diskovnih polj in s tem dodatno nižanje TCO.

Predlagano konfiguracijo povzema slika 4.3

4.5 Prilagoditve postopka za varovanje podatkov

Uveljavljena politika varnostnega kopiranja podatkov ne zadošča zahtevam za RPO. Za datotečne sisteme omogoča RPO en dan, za podatkovne baze pa eno uro. Za uskladitev z zahtevami bi morali inkrementalne kopije na datotečnih sistemih in kopiranje transakcijskih logov podatkovnih baz izvajati vsake pol ure, in vsebino "storage poolov" na strežniku za varnostno kopiranje podatkov pogosteje seliti na trakove, za kar bi potrebovali precej več tračnih pogonov v knjižnici. Tudi iznos trakov izven stavbe ima svoje pomanjkljivosti. Iznaša se v sosednjo stavbo, kar ne prepreči izgube podatkov v primeru katastrofe večjih razsežnosti (npr. potres, vojna, itd.). Iznos je ročni postopek, pri katerem skrbnik izvede določene ukaze, pobere trakove iz knjižnice in jih odnese. Ta postopek predstavlja varnostno tveganje.

Predlagamo zamenjavo tračne knjižnice z diskovno knjižnico in selitev tračne knjižnice na rezervno lokacijo. Tam se lahko uporabi kot shranjevalni prostor za drugo varnostno kopijo podatkov in na ta način izniči potrebo po iznosu trakov. Povezava med lokacijama je zasedena pretežno podnevi, v času največjega prirasta in sprememb podatkov. Ponoči je povezava bolj prosta in lahko služi za kopiranje podatkov iz diskovne knjižnice na primarni lokaciji na tračno knjižnico. Diskovna knjižnica z zmožnostjo definiranja navideznih tračnih pogonov omogoča lažje in hitrejše usklajevanje s potrebami politike varnostnega kopiranja podatkov.

4.6 Povzetek rešitve

Vodstvo fiktivnega naročnika želi zmanjšati stroške za hrambo podatkov in hkrati povečati njihovo razpoložljivost. Za zmanjševanje stroškov hrambe smo se odločili ponuditi centralizirano hrambo z uporabo strojne kompresije na diskovnih sistemih in vpeljavo hierarhičnega modela shranjevanja. Višjo razpoložljivost podatkov s predlagano rešitvijo

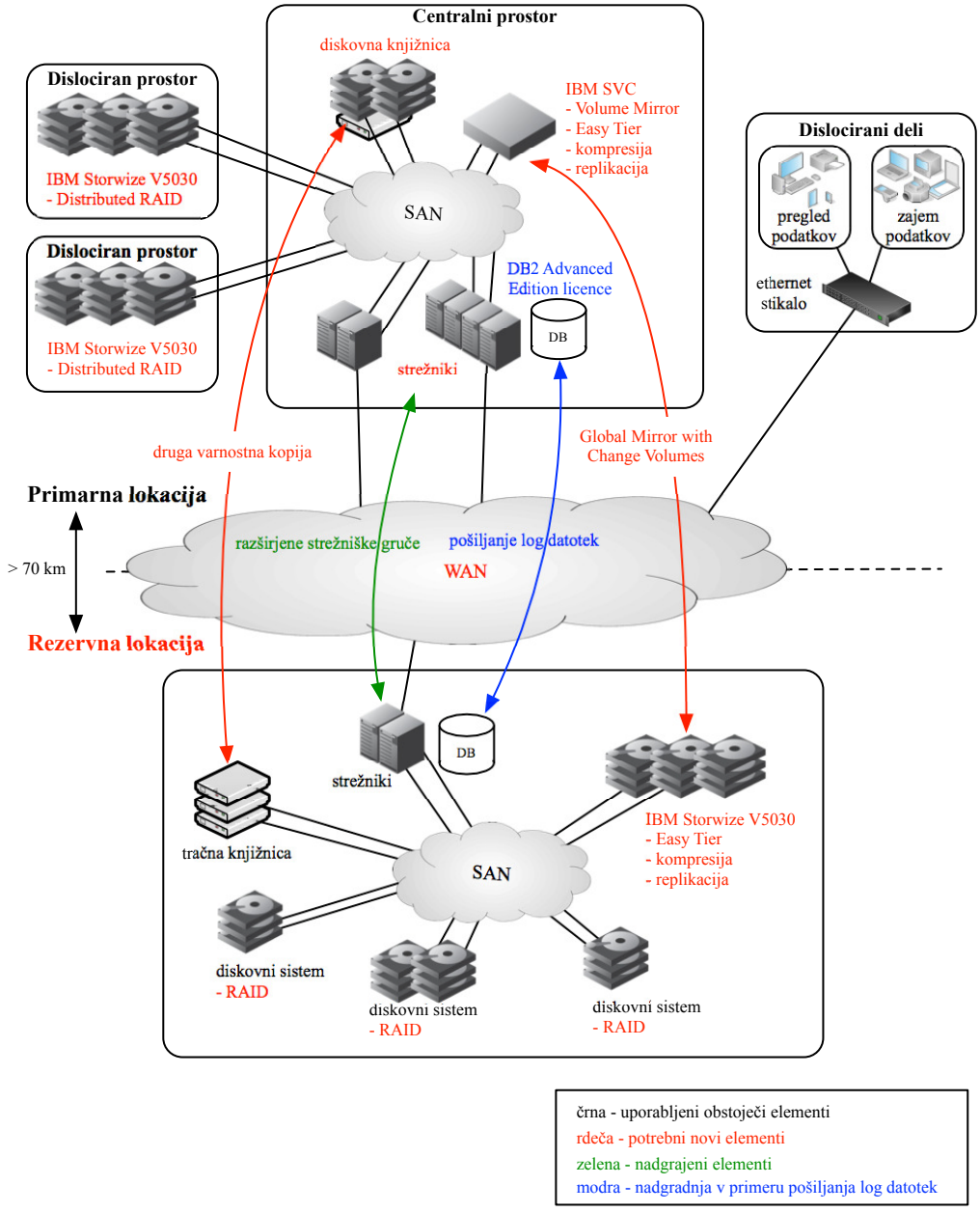
dosegamo na več nivojih. Pred okvarami na diskovnih sistemih varuje redundančna arhitektura. Za hitro okrevanje po napakah na diskih predlagamo uporabo distribuiranih RAID polj. Tudi programska oprema na diskovnem polju je lahko vzrok za težave, ki bi ga odpravili z zrcaljenjem podatkov na drug diskovni sistem na primarni lokaciji. Fiktivni naročnik ima zaradi distribuiranega informacijskaga sistema pripravljeno infrastrukturo, zato predlagamo postavitev diskovnih sistemov v ločene prostore in s tem zaščito pred lokalnimi kritičnimi dogodki. Rešitev predvideva zagotavljanje razpoložljivosti podatkov v primeru nesreče večjih razsežnosti z repliciranjem podatkov na rezervno lokacijo. Shema predlagane rešitve je predstavljena na sliki 4.4.

Opis korakov za implementacijo rešitve

Primarna lokacija:

- vzpostavimo nivo diskovne virtualizacije z IBM SVC,
- na dve lokaciji znotraj kampusa umestimo dva podrejena diskovna sistema IBM Storwize V5030,
- vse nove pomnilniške naprave povežemo v SAN omrežje po redundančnih poteh,
- diske na podrejenih diskovnih sistemih konfiguriramo v tri distribuirana RAID (DRAID) polja:
  - 1 x DRAID 4+P[2] z enim diskom za rezervo na 6 SSD diskih,
  - 5 x DRAID 7+P s štirimi diski za rezervo na 44 SAS diskih,
  - 7 x DRAID 7+P s štirimi diski za rezervo na 60 SATA diskih.
- na diskovni virtualizaciji vzpostavimo:
  - LUN-e za uporabo Easy Tier,
  - sinhrono zrcaljenje Volume Mirror med LUN-i iz podrejenih diskovnih sistemov,
  - strojno kompresijo,
  - asinhrono replikacijo podatkov na rezervno lokacijo z uporabo GMCV.
- varnostno kopiranje podatkov prestavimo iz tračne na diskovno knjižnico,
- podatke iz obstoječih diskovnih sistemov prepišemo skladno z zmožnostmi obstoječih rešitev tako, da so podatki čimbolj razpoložljivi.

[2] 4 diski za podatke, 1 disk za pariteto (P)



Slika 4.4 Shema predlagane rešitve.

Rezervna lokacija:

- vzpostavimo nivo diskovne virtualizacije z IBM Storwize V5030,
- ustreznim obstoječim diskovnim sistemom nadgradimo programsko opremo in jih uporabimo za pomnilniške kapacitete na rezervni lokaciji,
- vse pomnilniške naprave povežemo v SAN omrežje po redundančnih poteh,
- diske na diskovnih sistemih skonfiguriramo skladno z zmožnostmi diskovnih sistemov, (DRAID-5 na IBM Storwize, RAID-5 na ostalih)
- na diskovni virtualizaciji vzpostavimo:
  - LUN-e za uporabo Easy Tier[3],
  - strojno kompresijo,
  - replikacijo podatkov iz primarne lokacije.
- tračno knjižnico iz primarne prestavimo na rezervno lokacijo in jo uporabimo namesto iznosa trakov.

Če poslovni proces fiktivnega naročnika zahteva, po postavitvi obeh lokacij vzpostavimo še replikacijo med podatkovnimi bazami.

[3]Easy Tier na rezervni lokaciji bo bloke razvrščal drugače kot na primarni lokaciji

5 Zaključek

V diplomski nalogi je bil izveden pregled možnih tehnoloških rešitev za zagotavljanje razpoložljivosti podatkov in zmanjševanje stroškov njihovega hranjenja. S primerjavo dobrih in slabih lastnosti posameznih rešitev so bile izbrane optimalne rešitve glede na vstopne pogoje. Ugotovljeno je bilo, da:

- so nekatere privlačne rešitve ekonomsko smotrne šele nad določeno količino podatkov,
- je za doseganje nižjih TCO potrebno čimbolj enotno upravljanje,
- je potrebno za učinkovito rešitev izdelati načrt neprekinjenega poslovanja,
- je informacijski sistem lahko učinkovita podporna entiteta delovnim procesom le, če zadosti njihovim potrebam,
- so določene rešitve pogojene z geografskimi okoliščinami.

Rešitve za posamezno področje je pametno izrabljati samo na enem nivoju informacijskega sistema, sicer prihaja do nepotrebnega podvajanja porabe virov in s tem povezanimi višjimi stroški.

Potrebno je tudi preveriti skupno ceno za izvajanje določenih rešitev, saj npr. nakup drage licence zaradi varnostnih zahtev istočasno omogoči še kompresijo podatkov in vpeljavo "in-memory" modela podatkovnih baz, kar posledično pomeni zmanjšanje potrebne pasovne širine komunikacijskih povezav, manjšo potrebno količino diskovnega prostora in hkrati skrajšanje dostopnih časov in obdelav.

Zahteve po razpoložljivosti podatkov so zaradi vključenosti v projekt e-Zdravje pri vseh izvajalcih zdravstvene dejavnosti v Sloveniji podobne, rešitve pa so lahko zelo različne. Na rešitev vplivajo geografska lokacija izvajalca, količina pacientov, oddaljenost drugih zdravstvenih ustanov, itd.

Hkrati z zagotavljanjem razpoložljivosti podatkov bo potrebno poskrbeti tudi za časovno ustrezno hrambo podatkov skladno s pogoji, postavljenimi v Zakonu o zbirkah podatkov s področja zdravstvenega varstva[1]. Tukaj gre za občutljivo področje, ki je pri večini poslovnih subjektov, ki zbirajo informacije, še neobdelano. Podobno je tudi z Zakonom o varstvu osebnih podatkov[2], ki določa, pod kakšnimi pogoji je dovoljen vpogled v osebne podatke. Interoperabilna hrbtenica zagotavlja sledljivost dostopov do podatkov na certificiranih točkah izvajalcev, za ostale podatke pa je dolžan vsak izvajalec zdravstvene dejavnosti poskrbeti sam.

[1] http://www.uradni-list.si/1/objava.jsp?sop=2015-01-1933
[2] https://www.uradni-list.si/1/content?id=82668

Literatura

[1] Projekt e-Zdravje, [4. marec 2016] dostopno na:
http://www.ezdrav.si/category/projekti/

[2] Interoperabilnost, [4. marec 2016] dostopno na:
http://nio.ezdrav.si/?page_id=94

[3] Zakon o zdravstveni dejavnosti, 2. člen Uradni list RS št. 23/2005.

[4] Interoperabilna hrbtenica slovenskega zdravstva, [4. marec 2016] dostopno na:
http://www.sdmi.si/tl_files/pdf_materiali/3%20Interoperabilna%20hrbtenica.pdf

[5] Zdravstveno omrežje zNET, [4. marec 2016] dostopno na:
http://www.sdmi.si/tl_files/pdf_materiali/2%20%20zNET.pdf

[6] C. Brooks, M. Bedernjak, I. Juran, J. Merryman: Disaster Recovery Strategies with Tivoli Storage Management, ITSO, 2002.

[7] S. Nordell: Network latency - how low can you go?, [8. marec 2016]dostopno na:
http://www.lightwaveonline.com/articles/print/volume29/issue6/feature/network-latencyhowlowcanyougo.html

[8] J. Tate, S. Garraway, M. Hitchman: IBM Storwize V7000 and SANSlide Implementation, ITSO, 2013

[9] S. Kasampalis, Copy On Write Based File Systems Performance Analysis And Implementation, 2010, [21. marec 2016] dostopno na:
http://faif.objectis.net/download-copy-on-write-based-file-systems

[10] DB2 V10.1 Multi-temperature Data Management Recommendations, [1. april 2016] dostopno na:

http://public.dhe.ibm.com/software/dw/data/dm-1205multitemp/DB2V10_Multi-Temperature_0412.pdf

[11] J. Tate, M. Hart, H. Lonzer, T. J. Maluf, L. Miklas, J. Parkes, A. Saine, L. Sturmer, M. Tabinowski: Implementing the IBM Storwize V7000 and IBM Spectrum Virtualize V7.6, ITSO, 2015

[12] L. Vanel, R. van der Knaap, D. Foreman, K. Matsubara, A. Steel: AIX Logical Volume Manager, from A to Z: Introduction and Concepts, ITSO, 2000.

[13] K. Milberg: Optimizing AIX 5L performance: Tuning disk performance, Part 2, IBM, 2007, [14. marec 2016] dostopno na: http://www.ibm.com/developerworks/aix/library/au-aixoptimization-disktun2/

[14] IBM PowerHA Enterprise with GLVM, [14. marec 2016] dostopno na: http://www.ibm.com/support/knowledgecenter/#!/SSPHQG_7.2.0/com.ibm.powerha.geolvm/ha_glvmi_glvm.htm

[15] DRBD, [14. marec 2016] dostopno na: http://drbd.linbit.com/en/comp/drbd-linux-driver

[16] Aspera FASP high-speed transport, [14. marec 2016] dostopno na: http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?subtype=WH&infotype=SA&appname=SWGE_ZZ_MR_USEN&htmlfid=ZZW03306USEN&attachment=ZZW03306USEN.PDF

[17] Aspera FASP Software Environment Technology Capabilities, [14. marec 2016] dostopno na: http://asperasoft.com/fileadmin/media/Datasheets/fasp_Software_Environment_AsperaDS.pdf

[18] J. Tate, A. Bernasconi, A. Rainero, O. Rasmussen: IBM Storwize V7000, Spectrum Virtualize, HyperSwap, and VMware Implementation, ITSO, 2015

[19] IBM copy services, [10. marec 2016] dostopno na: https://www.ibm.com/support/knowledgecenter/#!/ST3FR7_7.6.1/com.ibm.storwize.v7000.761.doc/svc_copyservicesovr_21p99u.html

[20] C. Losinski, C. Stupca: Introduction to FlashCopy, [10. marec 2016] dostopno na: http://www.slideshare.net/HelpSystems/flash-copy-121213

[21] T. Pearson: Replication for Business Continuity, Disaster Recovery and High Availability, [11. marec 2016] dostopno na: http://www.slideshare.net/az990tony/replication-2013storageexpobrussels-upd3i?from_action=save

[22] J. Tate, R. Vilela Dias, I. Dikanarov, J. Kelly, P. Mescher: IBM System Storage SAN Volume Controller and Storwize V7000 Replication Family Services, ITSO, 2013

[23] J. Tate, C. Burns, D. Rubright, L. Sirett: IBM SAN Volume Controller and Storwize Family Native IP Replication, ITSO, 2014.

[24] Split mirror using suspended I/O in DB2 Universal Database, 2005, [17. marec 2016] dostopno na: http://www.ibm.com/developerworks/data/library/techarticle/dm-0508quazi/

[25] High availability disaster recovery (HADR), [17. marec 2016] dostopno na: http://www.ibm.com/support/knowledgecenter/#!/SSEPGG_10.5.0/com.ibm.db2.luw.admin.ha.doc/doc/c0011267.html

[26] DB2 pureScale Overview and Technology Deep Dive, IBM corporation, 2015, [17. marec 2016] dostopno na: http://slideplayer.com/slide/8949345/

[27] Data Guard Concepts and Administration, [17. marec 2016] dostopno na: https://docs.oracle.com/cd/B28359_01/server.111/b28294/concepts.htmi#i1033862

[28] Geographically dispersed DB2 pureScale cluster (GDPC), [17. marec 2016] dostopno na: https://www.ibm.com/support/knowledgecenter/#!/SSEPGG_10.5.0/com.ibm.db2.luw.licensing.doc/doc/c0060596.html

[29] File Compression and Decompression, [21. marec 2016] dostopno na: https://msdn.microsoft.com/en-us/library/windows/desktop/aa364219(v=vs.85).aspx

[30] Btrfs, [23. marec 2016] dostopno na: https://btrfs.wiki.kernel.org/index.php/Main_Page

[31] ZFS Administration, Part XI- Compression and Deduplication, 2012, [23. marec 2016] dostopno na:
https://pthree.org/2012/12/18/zfs-administration-part-xi-compression-and-deduplication/

[32] R. E. Sanders, T. Fanghaenel: Optimize storage with deep compression in DB2 10, IBM, 2012, [22. marec 2016] dostopno na:
http://www.ibm.com/developerworks/data/library/techarticle/dm-1205db210compression/

[33] C. Burns, B. Tuv-El, J. Quintal, J. Tate: IBM Real-time Compression in IBM SAN Volume Controller and IBM Storwize V7000, ITSO, 2015

[34] W. Tolliver: Do You Really Want Your Data Shingled?, [5. april] dostopno na:
https://shocksense.com/do-you-really-want-your-data-shingled/

[35] SNIA - Linear Tape File System (LTFS) Format Specification v2.3.0 rev 4 DRAFT, 2015, [30. marec 2016] dostopno na:
http://www.snia.org/sites/default/files/LTFS_Format_2 3 0_ver 0.4_150324b.pdf

[36] L. Coyne, K. Ngo, S. Neff: IBM Linear Tape File System Enterprise Edition V1.1.1.2: Installation and Configuration Guide, ITSO, 2015

[37] P. Feresten, L. Freeman, M. Woods: The NetApp Virtual Storage Tier, NetApp, 2011

[38] VNX FAST Cache, 2013, [23. marec 2016] dostopno na:
http://www.emc.com/collateral/software/white-papers/h8046-clariion-celerra-unified-fast-cache-wp.pdf

[39] Uprava RS za zaščito in reševanje: Ocena potresne ogroženosti Republike Slovenije, 2013, [5. april 2016] dostopno na:
http://www.sos112.si/slo/tdocs/ogrozenost_potres.pdf

[40] IBM Storwize V7000 Unified: Managing TSM integration, [9. april 2016] dostopno na:
http://www.ibm.com/support/knowledgecenter/#!/ST5Q4U/com.ibm.storwize.v7000.unified.143.doc/mng_tsm_topic_welcome.html